Aplicativos em PHP

De Wikibooks

Desenvolvimento de Aplicativos em PHP

For Linux e for Windows

Livro destinado ao desenvolvimento de aplicativos web em PHP.
Inicialmente era apenas uma idéia, agora já são mais de 600 páginas de bom conteúdo.
No início era só um colaborador, agora já são sete.
Traga sua colaboração e façamos algo juntos que seja grande e de muita utilidade para você e para todos os que lidam com desenvolvimento web.

1. Introdução
    1. História - 6
    2. Características e Recursos 10
2. InteligênciaEmocional 14
3. Instalação e configurações do ambiente
    1. Via Xampp 34
    2. Individualmente 41
4. Editores e IDEs para PHP, CSS, JavaScript e HTML
    1. Eclipse com PHPEclipse, Aptana e outros plugins . . . 47
    2. PHPDesigner 65
    3. Kate 66
    4. Dreamweaver 67
    5. Zend Studio 85
    6. VS.PHP 86
    7. Delphi for PHP 87
    8. KDevelop 89
    9. PSPAD 92
    10. Quanta 94

11.BlueFish 95
12.PHPEdit 96
13.Geany 97
14.NotePad++ 100
5. Referências
1. Windows 101
2. Linux 107
3. HTML 128
4. JavaScript 135
5. CSS 151
6. PHP 160
7. Livros e E-books 206
8. Cursos na área 208
6. Administração de SGBDs
1. MySQL 210
2. PostgreSQL 223
3. SQLite 244
7. Geradores de Aplicativos 248
1. phpCodeGenie (com MySQL) 252
2. SQLMaestro (com os principais SGBDs) 256
3. phpMyEdit 257
4. DadaBik 263
5. PHPLibDev 264
6. Web Form Generator 265
7. PHP Code Generator (com MySQL, PostgreSQL, Access ...) . 266
8. CMSs (Sistemas Gerenciadores de Conteúdo)
1. Joomla (Portais) 268
2. Mambo (Portais) 289
3. Drupal (Portais) 290
4. Xoops (Portais) 293
5. MediaWiki (wikis) 294
6. WordPress (blogs) 302
7. eGroupWare (colaboração) 304
8. Moodle (eLearning) 306

9. FrameWorks
    1. P4A 308
    2. Zend 323
    3. Seagull 324
    4. CodeIgniter 325
    5. QCodo 326
    6. Zephyr-php 328
    7. CakePHP 336
    8. Symfony 339
    9. Prado 342
    10. ZFRails 344
10. Abstrações de Bancos de Dados
    1. AdoDB 345
    2. PEAR/DB 362
11. Trabalhando em PHP com
    1. Formulários 376
    2. URLs 378
    3. Funções 380
    4. Constantes mágicas e super globais 389
    5. Session 392
    6. Arquivos e Diretórios 396
    7. Path 416
    8. Includes 419
    9. Formatação da saída 422
    10. Imagens e Gráficos 426
    11. Números 436
    12. Strings 438
    13. Permissões de arquivos e diretórios 446
    14. Validação de Dados 452
    15. Arrays 455
    16. XML 473
    17. Cookies 475
12. Integração do PHP com SGBDs
    1. MySQL 477

2. PostgreSQL 488
3. SQLite 495
4. IBM DB2 Express 496
13.Recursos Extras
1. Controle de Estoque 498
2. Loja Virtual 499
3. Código de Barras 502
4. Boleto bancário 512
5. PHP-GTK 513
6. Contadores de visitas 515
7. Livros de visitas 517
8. Chats 518
9. Suporte Online
10.Forums
11.Enquetes
12.Geradores de Relatórios 519
13.Integração com AJAX 531
14.Integração com Flash 535
15.Integração com Java 538
16.Bibliotecas de Funções 539
17.Tratamento de Erros
18.Compilando o PHP 542
19.Flex 543
14.Orientação a Objetos
1. Teoria e exercícios 549
2. Aplicativos de exemplo 562
3. Ferramentas 563
15.Aplicativos de Exemplo com
1. MySQL 565
2. PostgreSQL 566
3. SQLite 567
16.Apêndices
1. Análise e Projeto 568
2. Design 580

3. Hospedagem e Domínio 584
    1. Domínio 585
    2. Hospedagem de Site com cPanel 586
4. Segurança 609
5. Documentação 626
6. Informações Diversas 627
7. Oportunidade de Trabalho 629
    1. Elaboração de Currículos 634
    2. Técnicas de Entrevistas 637
    3. Preços de Serviços 638
8. Redes 640
9. Editores Gráficos 641

Contribuidores

- Daniel Ratkevicius (PHP+FLASH) - Contatos: daniel@ratkevicius.com.br[1] ou [2]
- Eric Silva - Contatos: eric_silva@msn.com[3]
- Haroldo Barbosa - Contatos: haroldob@gmail.com[4]
- Jackson Araujo - Contatos: jackson@designrio.com[5]
- Jamires Lontra [Varginha/MG] - Contato: jlontra@etdevarginha.com.br
- Ribamar FS - Contatos: ribafs[ ]users.sourceforge.net [6] e [7].
- William G. Comnisky - Contatos: w.comnisky@gmail.com[8] ou [9]

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP"

Aplicativos em PHP/Introdução/História

De Wikibooks

< Aplicativos em PHP | Introdução

Conteúdo

- 1 PHP na Wikipedia
- 2 A História do PHP
  - 2.1 Versão 1
  - 2.2 Versão 2
  - 2.3 Versão 3
  - 2.4 Versão 4
  - 2.5 Versão 5
  - 2.6 Versão 6

PHP na Wikipedia

A linguagem surgiu por volta de 1994, como um subconjunto de scripts Perl criados por Rasmus Lerdof, com o nome PHP/FI (Personal Home Page Tools/Forms Interpreter). Com as adições de Zeev Suraski e Andi Gutmans, dois programadores israelitas pertencentes ao Technion, o Instituto Israelita de Tecnologia, que reescreveram o parser, era lançada em 1997 a PHP 3, primeira versão estável e parecida com a linguagem atual. Ao reescrever o parser, foi criado o Zend Engine, que é mantido oficialmente pela empresa Zend em conjunto com a comunidade PHP. Em Maio de 2000 veio a público a versão 4, e em Julho de 2004, a versão 5, onde a principal mudança foi uma nova API para orientação a objetos provida pelo Zend Engine 2.

Trata-se de uma linguagem extremamente modularizada, o que a torna ideal para instalação e uso em servidores web. Diversos módulos são criados no repositório de extensões PECL (PHP Extension Community Library) e alguns destes módulos são introduzidos como padrão em novas versões da linguagem. É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem C e com a C++. Pode ser, dependendo da configuração do servidor, embutida no código HTML. Existem versões do PHP disponíveis para os seguintes sistemas operacionais: Windows, Linux, FreeBSD, Mac OS, OS/2, AS/400, Novell Netware, RISC OS, IRIX e Solaris

A Wikipédia funciona sobre um software inteiramente escrito em PHP, usando bases de dados MySQL: o MediaWiki.

Construir uma página dinâmica baseada em bases de dados é simples ,(em parte, vale lembrar), com PHP, este provê suporte a um grande número de bases de dados: Oracle, Sybase, PostgreSQL, InterBase, MySQL, SQLite, MSSQL, Firebird etc, podendo abstrair o banco com a biblioteca ADOdb, entre outras.

PHP tem suporte aos protocolos: IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC, SOAP. É possível abrir sockets e interagir com outros protocolos. E as bibliotecas de terceiros expandem ainda mais estas funcionalidades.

Existem iniciativas para utilizar o PHP como linguagem de programação de sistemas fixos (desktop). A mais notável é a PHP-GTK. Trata-se de um conjunto do PHP com a biblioteca GTK, portada do C++, fazendo assim softwares inter-operacionais entre Windows e Linux.

Segundo o Manual Oficial do PHP

A História do PHP e projetos relacionados

A História do PHP

O PHP percorreu um longo caminho ao nesses poucos anos. Criada para ser uma das mais poderosas e proeminentes linguagens da Web não foi uma tarefa fácil. Aqueles que se interessaram momentaneamente em ver como o PHP cresceu e se tornou o que é hoje, continue lendo. Versões do antigo PHP pode ser encontrado em » PHP Museum.

A História do PHP

Versão 1

PHP/FI

O PHP sucede de um produto mais antigo, chamado PHP/FI. PHP/FI foi criado por Rasmus Lerdorf em 1995, inicialmente como simples scripts Perl como estatísticas de acesso para seu currículo online. Ele nomeou esta série de script de 'Personal Home Page Tools'. Como mais funcionalidades foram requeridas, Rasmus escreveu uma implementação C muito maior, que era capaz de comunicar-se com base de dados, e possibilitava à usuários desenvolver simples aplicativos dinâmicos para Web. Rasmus resolveu » disponibilizar o código fonte do PHP/FI para que todos pudessem ver, e também usá-lo, bem como fixar bugs e melhorar o código.

PHP/FI, que significa Personal Home Page / Forms Interpreter, incluía algumas funcionalidades básicas do PHP que nós conhecemos hoje. Ele usava variáveis no estilo Perl, interpretação automática de variáveis vindas de formulário e sintaxe embutida no HTML. A sua própria sintaxe era similar a do Perl, porém muito mais limitada, simples, e um pouco inconsistente.

Versão 2

Em 1997, PHP/FI 2.0, a segunda versão da implementação C, obteve milhares de usuários ao redor do mundo (estimado), com aproximadamente 50,000 domínios reportando que tinha PHP/FI 2.0 instalado, angariando 1% dos domínios da Internet. Enquanto isto havia milhares de pessoas contribuindo com pequenos códigos para o projeto, e ainda assim

O PHP/FI 2.0 foi oficialmente lançado somente em Novembro de 1997, após perder a maior parte de sua vida em versões betas. Ele foi rapidamente substituído pelos alfas do PHP 3.0.

Versão 3

O PHP 3.0 foi a primeira versão que se assemelha ao PHP que nós conhecemos hoje. Ela foi criada por Andi Gutmans e Zeev Suraski em 1997 e foi totalmente reescrito, após eles descobrirem que o PHP/FI 2.0 poderia ajudá-los a desenvolver suas próprias aplicações de eCommerce de um projeto da Universidade. No esforço cooperativo e iniciativa de começar o PHP/FI à partir da base-usuário existente, Andi, Rasmus e Zeev decidiram cooperar e anunciar o PHP 3.0 como uma versão oficial de seu sucessor o PHP/FI 2.0, e o desenvolvimento do PHP/FI 2.0 foram descontinuados.

Uma das maiores características do PHP 3.0 era sua forte capacidade de extensibilidade. Além de oferecer aos usuários finais uma infraestrutura sólida para diversos bancos de dados, protocolos e APIs, o extensibilidade do PHP 3.0 atraí dezenas de desenvolvedores para se juntar e submeter novos módulos. Esta é a chave do tremendo sucesso do PHP 3.0. Outras características chaves introduzidas no PHP 3.0 foram o suporte à sintaxe para orientação à objetos e uma sintaxe muito mais poderosa e consistente.

Toda a nova versão da linguagem foi realizada sob um novo nome, que removeu a impressão do limitado uso pessoal que o PHP/FI 2.0 prendeu. Ela foi nomeada simplesmente 'PHP', com o significado que é um acrônimo - PHP: Hypertext Preprocessor.

No final de 1998, o PHP obteve uma base de dezenas de milhares de usuários (estimativa) e centenas de milhares de Web sites relatando que o tinham instalado. Em seu pico, o PHP 3.0 foi instalado em aproximadamente 10% dos servidores Web da Internet.

O PHP 3.0 foi oficialmente lançado em Junho de 1998, depois de ter passado aproximadamente 9 meses em testes públicos.

Versão 4

O PHP 3.0 foi oficialmente lançado em Junho de 1998, depois de ter passado aproximadamente 9 meses em testes públicos. Os objetivos do projeto eram melhorar a performance de aplicações complexas, e melhorar a modularidade do código base do PHP.Tais aplicações foram possíveis por causa das novas características do PHP 3.0 e o suporte a uma variedade de banco de dados de terceiros e APIs, mas o PHP 3.0 não foi projetado para trabalhar com aplicações muito complexas eficientemente.

A nova engine, dublado 'Zend Engine' (conhecidos pelos seus primeiros nomes, Zeev e Andi), fazendo desse objetivo um sucesso, e foi introduzida em meados de 1999. PHP 4.0, baseado nesta engine, e acompanhado com uma série de novas características, foi oficialmente lançada em Maio de 2000, quase dois anos após o seu predecessor, o PHP 3.0. Além da altíssimo melhoramento da performance desta versão, o PHP 4.0 incluiu outras características chave como o suporte para muitos servidores WEB, sessões HTTP, buffer de saída, maneiras mais seguras de manipular input de usuários e muitas construções novas na linguagem.

PHP 4 é a última versão lançada do PHP. O trabalho já começou na modificação e melhoramento da Zend Engine para integrar novas características que foram designadas para o PHP 5.0.

Hoje, o PHP está começando a ser usado por centenas de milhares de desenvolvedores (estimativa), e muitos milhões de sites reportam que tem o PHP instalado, que explica os 20% de domínios da Internet.

A equipe de desenvolvimento do PHP contém dezenas de desenvolvedores, bem como dezenas de outros que trabalham com projetos relacionados ao PHP como o PEAR e a documentação do projeto.

Versão 5

A última versão 5 estável do PHP é a 5.2.3

Download em http://www.php.net/downloads.php#v5

Notas da versão http://www.php.net/releases/5_2_3.php

Changelog http://www.php.net/ChangeLog-5.php#5.2.3

Versão 6

- Primeiras informações do PHP 6.0 - http://www.robsonjunior.com.br/post/primeiras-informacoes-php6/

- O PHP6 ainda não saiu mas já está causando polêmicas. Veja os artigos:

Prepare for PHP 6 - http://www.corephp.co.uk/archives/19-Prepare-for-PHP-6.html

- Mudanças no PHP6 são polêmicas - http://www.meiobit.com/destaque/mudancas_no_php6_serao_polemicas

- PHP6 - O PRINCIPIO DA MUDANÇA - http://phpexperts.blogspot.com/2007/03/php6-o-principio-da-mudana.html

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Introdu%C3%A7%C3%A3o/Hist%C3%B3ria"
Page categories: PHP

Aplicativos em PHP/Introdução/Características e Recursos

De Wikibooks

< Aplicativos em PHP | Introdução

Segundo a Wikipedia em http://pt.wikipedia.org/wiki/Php#Hist.C3.B3rico

A linguagem PHP é uma linguagem de programação de domínio específico, ou seja, seu escopo se concentra em um campo de atuação que é o Desenvolvimento Web, embora tenha variantes como o PHP-GTK.

Seu propósito é solucionar problemas Webs da maneira mais fácil possível.

Conteúdo

- 1 Características principais:
- 2 O Que o PHP Pode Fazer?
- 3 Esses são os maiores campos onde os scripts PHP podem ser utilizados:
- 4 Suporte a Sistemas Operacionais
- 5 Suporte a Flash, PDF e XML
- 6 Suporte a SGBDs
- 7 Diversos Outros Recursos

Características principais:

```
* Orientação a objeto
* Portabilidade
* Independência de plataforma - "write once run anywhere";
* Tipagem fraca
* Sintaxe similar a Linguagem C/C++ e o PERL
```

Segundo o Manual oficial

O Que o PHP Pode Fazer?

Qualquer coisa. O PHP é focado para ser uma linguagem de script do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer, como: coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas o PHP pode fazer muito mais.

Esses são os maiores campos onde os scripts PHP podem ser utilizados:

- Script no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para seu trabalho. O interpretador do PHP (como CGI ou módulo), um servidor web e um browser. Basta rodar o servidor web conectado a um PHP instalado. Você pode acessar os resultados de seu programa PHP com um browser, visualizando a página PHP através do servidor web. Todos eles podem rodar na sua máquina, em casa, para você experimentar programação com o PHP. Veja a seção das instruções de instalação para mais informações. - Script de linha de comando. Você pode fazer um script PHP funcionar sem um servidor web ou browser. A única coisa necessária é o interpretador. Esse tipo de uso é ideal para script executados usando o cron ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto. Veja a seção Utilizando o PHP em linha de comando para maiores informações. - Escrevendo aplicações desktop. O PHP provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas se você conhece bem o PHP, e gostaria de usar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, você pode usar o PHP-GTK para escrever programas assim. Isso ainda lhe habilita a escrever aplicações multi-plataformas. O PHP-GTK é uma extensão do PHP, não disponibilizada na distribuição oficial. Caso esteja interessado no PHP-GTK, visite o site do projeto (http://www.gtk.org/).

Suporte a Sistemas Operacionais

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes Unix (incluindo HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS, e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atuais, incluindo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet Servers, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd, e muitos outros. O PHP pode ser configurado como módulo para a maioria dos servidores, e para os outros como um CGI comum.

Com o PHP, portanto, você tem a liberdade para escolher o sistema operacional e o servidor web. Do mesmo modo, você pode escolher entre utilizar programação estrutural ou programação orientada a objeto, ou ainda uma mistura deles. Mesmo sem todos os recursos da POO (Programação Orientada a Objetos) implementados no PHP 4, muitas bibliotecas de código e grandes aplicações (incluindo a biblioteca PEAR) são escritas somente em código POO. O PHP 5 corrige as fraquezas da POO do PHP 4, e introduz um modelo de objetos completo.

Suporte a Flash, PDF e XML

Com PHP você não está limitado a gerar somente HTML. As habilidades do PHP incluem geração de imagens, arquivos PDF e animações Flash (utilizando libswf ou Ming) criados dinamicamente, on the fly. Você pode facilmente criar qualquer padrão texto, como XHTML e outros arquivos XML. O PHP pode gerar esses padrões e os salvar no sistema de arquivos, em vez de imprimi-los, formando um cache dinâmico de suas informações no lado do servidor.

Suporte a SGBDs

Talvez a mais forte e mais significativa característica do PHP é seu suporte a uma ampla variedade de SGBDs. Escrever uma página que consulte um banco de dados é incrivelmente simples. Os seguintes SGBDs são atualmente suportados:

```
Adabas D         InterBase       PostgreSQL
dBase         FrontBase       SQLite
Empress       mSQL    Solid
FilePro (read-only)   Direct MS-SQL   Sybase
Hyperwave     MySQL   Velocis
IBM DB2       ODBC    Unix dbm
Informix      Oracle (OCI7 and OCI8)
Ingres        Ovrimos
```

Também foi providenciada uma abstração de banco de dados DBX permitindo a você utilizar qualquer SGBD transparentemente com sua extensão. Adicionalmente, o PHP suporta ODBC (Open Database Connection, ou Padrão Aberto de Conexão com Bancos de Dados), permitindo que você utilize qualquer outro SGBD que suporte esse padrão mundial.

Diversos Outros Recursos

O PHP também tem suporte para comunicação com outros serviços utilizando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows) e incontáveis outros. Você pode abrir sockets de rede e interagir diretamente com qualquer protocolo. O PHP também suporta o intercâmbio de dados complexos WDDX, utilizado em virtualmente todas as linguagens de programação para web. Falando de comunicação, o PHP implementa a instanciação de objetos Java e os utiliza transparentemente como objetos PHP. Você ainda pode usar sua extensão CORBA para acessar objetos remotos.

O PHP é extremamente útil em recursos de processamento de texto, do POSIX Estendido ou expressões regulares Perl até como interpretador para documentos XML. No processamento de XML, o PHP 4 suporta os padrões SAX e DOM, além de você também poder utilizar a extensão XSL para transformar documentos XML. O PHP 5 padroniza toda a extensão XML a partir da base sólida da libxml2, além de estender os recursos com o acréscimo ao SimpleXML e XMLReader.

Utilizando o PHP no campo do e-commerce, você poderá usar as funções específicas para Cybescash, CyberMUT, Verysign Payflow Pro e MCVE, práticos sistemas de pagamento online.

Por último mas longe de terminar, temos também outras extensões interessantes: funções para o search engine mnoGoSearch, funções para Gateway IRC, vários utilitários de compressão (gzip, bz2), calendário e conversões de datas, tradução...

Como você pode ver, esta página não é suficiente para descrever todos os recursos e benefícios que o PHP pode oferecer. Leia nas seções sobre a Instalação do PHP, e veja a referência das funções para detalhes das extensões mencionadas aqui.

Popularidade do PHP

Atualmente o PHP é uma das mais populares linguagens de desenvolvimento da Web, provavelmente a mais.Veja as estatísticas neste site: http://www.nexen.net/chiffres_cles/phpversion/.

Vale lembrar que ser a mais popular gera muitas vantagens mas também gera desvantagens. Não é bom ficar se iludindo que apenas por ser a mais popular é a melhor para tudo, pois não é verdade. PHP é uma ótima linguagem, fácil de aprender, conta com muitos recursos já prontos para o programador, tem uma ótima documentação e uma grande comunidade ao redor do mundo, mas está longe de ser perfeita. Sei também que nenhuma é perfeita, mas lembre-se que para que você possa fazer uma avaliação adequada de que é realmente necessário para o que você precisa, deverá conhecer bem a linguagem para só então fazer uma avaliação correta.

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Introdu%C3%A7%C3%A3o/Caracter%C3%ADsticas_e_Recursos"
Page categories: PHP

Aplicativos em PHP/InteligênciaEmocional

De Wikibooks

< Aplicativos em PHP

Conteúdo

- 1 O Que PHP tem a ver com Inteligência Emocional?
- 2 Ser Inteligente de Fato
- 3 Falta de Inteligência Emocional
- 4 Valorização do Profissional Inteligente Emocionalmente
- 5 Algumas Idéias do Daniel Goleman
- 6 Como Melhorar a IE?
- 7 Inteligência Emocional na Wikipedia
- 8 Administrando com Inteligência Emocional
- 9 Concluindo sem fechar
- 10 Resumo do Livro Caindo na Real
- 11 Dicas para ser mais criativo
- 12 Regras de Ouro para uma boa Estimulação
- 13 Referências

O Que PHP tem a ver com Inteligência Emocional?

"Até pouco tempo atrás o sucesso de uma pessoa era avaliado pelo raciocínio lógico e habilidades matemáticas e espaciais (QI). Mas o psicólogo Daniel Goleman, PhD, com seu livro "Inteligência Emocional" retoma uma nova discussão sobre o assunto. Ele traz o conceito da inteligência emocional como maior responsável pelo sucesso ou insucesso das pessoas. A maioria das situações de trabalho é envolvida por relacionamentos entre as pessoas. Desta forma pessoas com qualidades de relacionamento humano, como afabilidade, compreensão, gentileza têm mais chances de obter o sucesso."

Essa imagem representa uma forma sadia de crescer na vida e profissionalmente: subir ajudando outros a subirem. Isso mostra generosidade e sabedoria. Imagine alguém que cresce assim: como será o comportamento do colega que ele ajudou a crescer? Será que ele irá procurar prejudicá-lo em sua carreira.

Agora visualizemos (nem mostrarei imagem :)) alguém que sobe pisando em quem estiver em seu caminho. Será essa a forma mais sábia de subir? Será que os colegas machucados em seu caminho irão ajudá-lo a continuar subindo?

Acontece que todo programador é um ser humano, com sentimentos e emoções como qualquer outro. É importante lembrar que esse lado do programador é muito importante em seu desempenho final e dependendo de como lidarmos com nossas emoções poderemos ser mais ou menos bem sucedidos em nossos empreendimentos.

Geralmente qualquer especialista considera-se superior aos demais por ter um bom domínio de uma área que os demais não conhecem. Isso o leva a menosprezar outros conhecimentos que não sejam os de sua área. Com isso fecha-se deixando de aprender muitas coisas importantes e isola-se em seus relacionamentos.

Ser Inteligente de Fato

Para sermos realmente inteligentes devemos ser sensíveis a todas as áreas de atuação humano e até além delas (já que existem áreas que estão além dos conhecimentos, como a criatividade, a estética, etc). Particularmente a área psicológica deve ser levada a sério. Em nossa cultura praticamente não se fala nem se estuda sobre nosso interior, nossas emoções e cia.

Falta de Inteligência Emocional

De que adianta ter um grande potencial, ter muito conhecimento, ser muito inteligente e criativo em algumas áreas e não conseguir conviver bem com suas emoções e sair criando confusão a todo o momento com clientes, colegas de trabalho, familiares, etc? Para realmente usufruir do nosso potencial e ainda por cima melhorá-lo a cada dia devemos aprender a lidar da melhor maneira possível com nossas emoções e sentimentos.

Valorização do Profissional Inteligente Emocionalmente

Atualmente as grandes e organizadas empresas estão dando muita importância para algo que se está chamando de Inteligência Emocional, que é esta capacidade de lidar bem com o lado emocional, especialmente para os cargos mais importantes das empresas.

Algumas Idéias do Daniel Goleman

Existe polêmica sobre quem é o pai da inteligência emocional mas quanto ao seu principal divulgador não existe dúvida, é o psicólogo Daniel Goleman.

Algumas habilidades emocionais são consideradas importantes para que uma pessoa alcance seus objetivos, seja feliz e alcance sucesso na vida. Dentre elas são citadas o controle do temperamento, adaptabilidade, persistência, amizade, respeito, amabilidade e empatia. Goleman apresenta os seguintes níveis de Inteligência Emocional:

- Auto-conhecimento emocional - Autoconsciência: conhecimento que o ser humano tem de si próprio, de seus sentimentos ou intuição. Esta competência é fundamental para que o homem tenha confiança em si (autoconfiança) e conheça seus pontos fortes e fracos;

- Controle emocional - Capacidade de gerenciar os sentimentos: é importante saber lidar com os sentimentos. A pessoa que sabe controlar seus próprios sentimentos se dá bem em qualquer lugar que esteja ou em qualquer ato que realize.

- Auto- motivação - Ter vontade de realizar, otimismo: Pôr as emoções a serviço de uma meta. A pessoa otimista consegue realizar tudo que planeja pois tem consciência que todos os problemas são contornáveis e resolvíveis.

- Reconhecer emoções nos outros - Empatia: saber se colocar no lugar do outro. Perceber o outro. Captar o sentimento do outro. A calma é fundamental para que isso aconteça. Os problemas devem ser resolvidos através de conversas claras. As explosões devem ser evitadas para que não prejudique o relacionamento com os outros.

- Habilidade em relacionamentos inter-pessoais - Aptidão social: a capacidade que a pessoa deve ter para lidar com emoções do grupo. A arte dos relacionamentos deve-se, em grande parte em saber lidar com as emoções do outro. Saber trabalhar em equipe é fundamental no mundo atual.

Os sentimentos mais fortes do homem são a tristeza, a alegria e a raiva. É fundamental saber lidar com eles. As pessoas que sabem controlar suas emoções são aquelas que obtém mais sucesso na vida, em qualquer tipo de medição, provas de vestibular, etc.

Segundo Goleman: "emoções são sentimentos a se expressarem em impulsos e numa vasta gama de intensidade, gerando idéias, condutas, ações e reações. Quando burilados, equilibrados e bem-conduzidos transformam-se em sentimentos elevados, sublimados, tornando-se, aí sim – virtudes."

Como Melhorar a IE?

Vale ressaltar que não ajuda ficar dizendo ou pensando: "Eu sou assim mesmo e não posso mudar". O primeiro passo é perceber e depois admitir que existe algo em você que precisa melhorar. Depois disso é procurar se informar ao máximo sobre o assunto e ter consciência de que pode e deve fazer algo para mudar. Fique atento, observe a si mesmo nas relações, procure perceber por que age e reage assim. Sempre que perceber uma reação que deseja mudar, pare, perceba e procure mudar. Não condene a si por ser como é, isso de nada ajuda. Encare os fatos:existe algo em você que pretende mudar e é isso que irá fazer. Não desanime com as dificuldades, lembre que está fazendo algo que o tornará melhor, melhor profissional, mais valorizado, mais importante para os colegas, para a empresa e para você também. Alguém melhor.

Inteligência Emocional na Wikipedia

Na psicologia, inteligência emocional é um tipo de inteligência que envolve as habilidades para perceber, entender e influenciar as emoções. Foi introduzida e definida por John D. Mayer e Peter Salovey. Inteligência emocional, chamada também EI é medida frequentemente como um Quociente de inteligência Emocional ou um QE emocional, descrevem uma habilidade, uma capacidade, ou uma habilidade de perceber, para avaliar e controlar as emoções de si mesmo, de outro, e dos grupos. Entretanto, sendo uma área relativamente nova, a definição da inteligência emocional está ainda em um estado do fluxo. Alguns, tais como John D. Mayer (2005a) preferem distinguir o conhecimento emocional da inteligência emocional, como discutido abaixo.

Administrando a IE

Em empresas ultrapassadas, o executivo que não se deixava levar pelo lado emocional em relação aos seus funcionários era valorizado, pela simples crença de que seria impossível lidar com as pessoas em situações que exigissem uma atitude mais drástica. Pensava-se que usar o coração no lugar da cabeça iria causar conflitos entre interesses pessoais e as metas organizacionais. As decisões duras não poderiam ser tomadas sem um distanciamento afetivo, mesmo havendo a probabilidade de tomá-las de modo mais humano. Entretanto, esse tipo de profissional é, claramente, do passado. **O executivo atual é aquele que possui uma elevada habilidade de relacionamento inter-pessoal**. **Um administrador mais humano é extremamente necessário nos dias atuais e futuramente ainda mais**. **O custo-benefício da utilização da IE nas empresas é uma idéia nova, porém muito evidente. O administrador deve proporcionar abertura para sugestões, saber criticar pela melhor forma, assim como elogiar, deve também saber como dissuadir preconceitos e trabalhar em equipe. O ambiente de trabalho não deve ser um ambiente de discórdia, intriga, sem abertura e amedrontador.**

Administrando com Inteligência Emocional

Começarei pelo 1º fator importante na administração da IE:

- **O feedback** Como é o melhor meio de se aplicar o feedback? O feedback não pode ser postergado. Uma vez que ele não é efetuado enquanto o assunto está "quente", ele acaba não sendo mais tão específico, fica acumulado e acaba sendo aplicado de forma ampla e destrutiva.

- **Ser específico**. É muito importante utilizar a especificidade tanto na crítica, como no elogio. O foco deve ser somente no problema e não no trabalho inteiro.

- **Oferecer uma solução**. A crítica deve sempre ser acompanhada de uma sugestão de como melhorar. Sem uma solução, a crítica fica vazia e acaba atingindo o indivíduo pessoalmente, deixando-o desmoralizado.

- **Fazer a crítica pessoalmente**. Não deve-se criticar ou elogiar por outro meio que não seja cara a cara. Este é o meio mais franco de se passar alguma informação, além de estar fornecendo a oportunidade de esclarecimentos.

- **Seja sensível**. Tente ser empático com seus comandados, você deve possuir uma sintonia emocional com as pessoas, sinta o que elas precisam, o que querem dizer, ou seja, o que elas estão sentindo.

Concluindo sem fechar

Lembre que você vive no mundo em constante relacionamento com outros seres humanos que poderão te ajudar ou te prejudicar, dependendo muito de como você os trata. Não basta apenas ajudar com a intenção de ser ajudado ou de evitar ser prejudicado, é importante que realmente sinta a vontade de colaborar, de melhorar a vida de outras pessoas. Mesmo quando estiver pretendendo tirar proveito, obter lucro, mesmo nestas situações, tenha em mente que o seu serviço ou produto irá ajudar a melhorar a vida de outras pessoas. Ganhar dinheiro não deve ser o objetivo principal, mas uma conseqüência de um trabalho, ou serviço, bem feito e que de fato ajuda outras pessoas.

Resumo do Livro Caindo na Real

Apenas clique no linque acima para ler o resumo.

Dicas para ser mais criativo

- Nunca se contente com a primeira idéia que lhe ocorrer. Busque outras para, entre muitas, escolher a melhor

- Não se acomode. Sempre existe uma maneira de fazer melhor, mais rápido ou com menor custo aquilo que você já faz. Se você não pensar nisso, alguém irá pensar

- Seja curioso. Evite reproduzir tarefas mecanicamente. Busque as causas, os porquês, as implicações. Muitas idéias surgem daí

- Idéias não saem do nada. Associe, adapte, substitua, modifique, reduza. As combinações são infinitas

- Não acredite em bordões como "isso nunca vai funcionar" ou "em time que está ganhando não se mexe". O novo sempre assusta. Toda idéia tem de quebrar resistências

- Tenha iniciativa. Muitas boas idéias acabam no fundo da gaveta porque seus autores não tomam a decisão de mostrá-las aos outros

- Ouça os outros. Principalmente se eles pensam diferente de você. As idéias se desenvolvem com a divergência

- Faça de vez em quando coisas que contrariem seus hábitos, no trabalho ou no lazer. Por exemplo: se você gosta de filmes de ação, assista a um drama romântico. Se é fã de rock, tente o jazz. Sair da rotina é sempre estimulante para o cérebro

---

Esta mensagem pode ser encontrada no site "Contando Histórias", no endereço http://www.contandohistorias.com.br/historias/2004097.php

---

Veja também:

-- Um aluno diferente -- [http://www.contandohistorias.com.br/historias/2006195.php]

-- O humor cura -- [http://www.contandohistorias.com.br/historias/2006089.php]

-- A cobra e o vagalume -- [http://www.contandohistorias.com.br/historias/2004007.php]

-- As estatísticas do amor -- [http://www.contandohistorias.com.br/historias/2004047.php]

-- Amizade -- [http://www.contandohistorias.com.br/historias/2004028.php]

Regras de Ouro para uma boa Estimulação

Do site Portal da Família - http://www.portaldafamilia.org/artigos/artigo161d.shtml

Recomendações para pais mas boa parte se aplica a praticamente todas os casos.

A mente bem desenvolvida, a paixão por aprender e a habilidade de fazer o conhecimento funcionar são as novas chaves para o Futuro.

```
Regras de ouro para uma boa estimulação

 - Deve ser nova;
 - Interessante;
 - Alegre;
 - Variada;
 - Curta, que não cansa;
 - Lúdica: seu objetivo é brincar aprendendo;
 - Do mais fácil ao mais complicado;
 - Empregando uma pedagogia positiva, reforçando seus êxitos;
 - O melhor presente é um sorriso de seus pais ou uma carícia;
 - Sem perguntas de nenhum tipo – não se fazem exames;
 - Os "melhores jogos" de estimulação são vocês mesmos.

 * Para aprofundar no assunto, procure ler:

 - "Como multiplicar a inteligência do seu bebê" – Glenn Doman
 - "Como ensinar seu bebê a ler" – Glenn Doman
 - "Como ensinar matemática a seu bebê" – Glenn Doman
     Editora Artes e Ofícios
```

Referências

- Bom site com artigos e links sobre Inteligência Emocional - http://www.din.uem.br/ia/emocional/

- Teste de Inteligência Emocional - http://www.cpsimoes.net/testeqie/testeqie.html

- Artigos sobre Inteligência Emocional - http://www.seducaoecomportamento.com.br/autodes06.htm

- Inteligência Emocional na Sala de Aula - http://www.centrorefeducacional.com.br/intemoci.htm

- Entrevista com Daniel Goleman - http://www.abrae.com.br/entrevistas/entr_gol.htm

- Um Estudo de Validade sobre a Capacidade de Perceber Emoções - http://www.scielo.br/pdf/prc/v16n2/a08v16n2.pdf

- AUTODIAGNÓSTICO: TESTE SUA INTELIGÊNCIA EMOCIONAL - http://www.guiarh.com.br/z84.htm

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Intelig%C3%AAnciaEmocional"
Page categories: PHP

Aplicativos em PHP/InteligênciaEmocional/Resumo do Livro Caindo na Real

De Wikibooks

< Aplicativos em PHP | InteligênciaEmocional

2.8 - Resumo do Livro "Caindo na Real"

O que é Caindo na Real?

Quer construir uma aplicação web de sucesso? Então é hora de Cair na Real. Caindo na Real é o menor, mais rápido e melhor caminho para construir software.

- Caindo na Real é sobre pular todas as coisas que 'não' * representam a realidade (cartas, gráficos, caixas, setas, esquemas, wireframes, etc.) e **realmente construir a coisa real**.

- Caindo na Real é menos. Menos massa, menos software, menos funcionalidades, menos papéis, **menos tudo que não é essencial** (e a maioria do que você pensa ser essencial realmente não é).

- Caindo na Real é **permanecer pequeno e ser ágil**.

- Caindo na Real **inicia com a construção da interface**, ou seja, as telas reais que as pessoas irão utilizar. Começa com as experiências reais dos clientes, construindo a partir disso para trás. Dessa forma você obtém a interface adequada antes de obter um software errado.

- Caindo na Real é sobre **iterações e baixar os custos da mudança**. Caindo na Real tem tudo a ver com lançamento, refinamento e melhorar constantemente, o que o torna o caminho perfeito para software baseado em web.

- Caindo na Real **entrega exatamente o que os clientes precisam** e elimina qualquer coisa que não precisam.

* adicionei por considerar que faltava

Os benefícios de Caindo na Real

Caindo na Real entrega melhores resultados porque o força a lidar com os problemas reais que está tentando resolver em vez de suas idéias sobre esses problemas. Ele o força a lidar com a realidade.

Caindo na Real pula especificações funcionais e outras documentações transitórias em favor de construir telas reais. Uma especificação funcional é para inglês ver, uma ilusão de um acordo, enquanto uma página web pronta é realidade. É isso que seus clientes irão ver e usar. É isso que importa. Caindo na Real o leva lá mais rápido. E isso signfica que está tomando decisões de software baseado na coisa real em vez de noções abstratas.

Finalmente, Caindo na Real é a maneira que se encaixa idealmente para software baseado em web. O modelo convencional de entregar software em uma caixa e então esperar um ano ou dois para entregar uma atualização está desaparecendo. Diferente de software instalado, aplicações web podem evoluir constantemente de maneira diária. Caindo na Real abre essa vantagem por tudo que ele vale.

Como Escrever Software Vigoroso

Escrita vigorosa é concisa. Uma sentença não deve conter palavras desnecessárias, um parágrafo não deve conter sentenças desnecessárias, pela mesma razão que desenhar não deve ter linhas desnecessárias e uma máquina não deve ter partes desnecessárias. Isso requer não que o escritor torne todas as sentenças curtas ou evite todos os detalhes e trate os assuntos apenas em ítens, mas sim que cada palavra fale. --De "Os Elementos de Estilo" de William Strunk Jr.

Números de versão? Jogue pela janela. Você precisa construir, lançar e refinar. Então recomece e repita.

Acreditamos que software é muito complexo. Funcionalidades demais, botões demais, coisa demais para aprender. Nossos produtos fazem menos do que a concorrência -- intencionalmente. Construímos produtos que funcionam de forma mais esperta, que parecem melhor, que lhe permitem fazer suas coisas e são mais fáceis de usar.

O primeiro passo é quebrar em pequenas unidades. Quando existem pessoas demais envolvidas, nada acontece. Quanto mais enxuto você for, mais rápido – e melhor – as coisas acontecem.

Lance menos funcionalidades, mas de qualidade. Você não precisa usar a forma big bang com todo novo lançamento e amontoados de funcionalidades. Dê aos usuários pedaços minúsculos que eles possam digerir.

Construa software para você mesmo

Uma grande maneira de escrever software é começar resolvendo seus próprios problemas. Você será o público-alvo e saberá o que é importante e o que não é. Isso lhe dá um bom adiantamento na entrega de um produto fora de série.

A chave aqui é entender que não está sozinho. Se estiver tendo problemas, é provável que centenas de milhares de outras pessoas estão no mesmo barco. Esse é seu mercado. Não foi fácil?

Basecamp se originou em um problema: como uma empresa de design precisávamos de uma maneira simples de comunicar nossos clientes sobre os projetos. Começamos fazendo isso através da extranet dos clientes, que atualizávamos manualmente. Mas modificar o HTML na mão toda vez que o projeto precisava ser atualizado simplesmente não estava funcionando. Esses sites de projetos sempre pareciam ficar travados e eventualmente eram abandonados. Era frustrante porque nos deixava desorganizados e deixava os clientes no escuro.

Então começamos a procurar outras opções. Ainda assim cada ferramenta que encontrávamos ou 1) não fazia o que precisávamos ou 2) era gorda de funcionalidades que não precisávamos – como cobrança, controles estritos de acesso, planilhas, gráficos, etc. Sabíamos que deveria haver uma maneira melhor então decidimos construir nossa própria.

Quando resolvemos nossos próprios problemas, criamos uma ferramenta que nos apaixona. E paixão é a chave. Paixão significa que realmente a usaremos e cuidaremos dela. E essa é a melhor maneira de fazer os outros se sentirem apaixonados sobre ela também. Arranhando sua própria coceira

O mundo de Código Aberto abraçou esse mantra há muito tempo – eles chamam de "arranhando sua própria coceira". Para os desenvolvedores de código aberto, significa que terão as ferramentas que querem, entregues da maneira que querem. Mas os benefícios vão mais a fundo.

Como designer ou desenvolvedor de uma nova aplicação, você precisa encarar centenas de micro-decisões todos os dias: azul ou verde? Uma tabela ou duas? Estática ou dinâmica? Abortar ou recuperar? Como tomamos essas decisões? Se é algo que reconhecemos como importante, poderíamos perguntar. O resto, chutamos. E todos esses chutes constroem um tipo de débito em nossas aplicações – uma rede interconectada de coisas que assumimos.

Como um desenvolvedor, detesto isso. O conhecimento de todas essas bombas-relógio em pequena escala nas aplicações que escrevo somam-se ao meu stress. Desenvolvedores de código aberto, arranhando suas próprias coceiras, não sofrem isso. Porque eles são seus próprios usuários, eles sabem a resposta correta para 90% das decisões que precisam tomar. Acho que é uma das razões que as pessoas chegam em casa após um dia duro de trabalho de codificação e ainda trabalham com código aberto: é relaxante.

— Dave Thomas, The Pragmatic Programmers

Você precisa de importar sobre isso

Quando você escreve um livro, precisa de mais do que uma história interessante. Precisa ter um desejo de contar a história. Precisa investir pessoalmente de alguma maneira. Se vai viver com alguma coisa por dois anos, três anos, o resto de sua vida, precisa se importar sobre isso. ""Precisa sentir paixão"". —Malcolm Gladwell, autor (de Algumas Finas Fatias de Malcolm Gladwell)

Dinheiro de fora é plano B

A primeira prioridade de muitas empresas iniciantes é adquirir fundos de investidores. Mas lembre-se, se nos viramos para gente de fora para fundos, teremos que responder a eles também. Crescem expectativas. Investidores querem seu dinheiro de volta – e rapidamente. O fato triste é que dinheiro entrando nem sempre significa a construção de um produto de qualidade.

Atualmente não é preciso muito para começar. Hardware é barato e uma boa parte de grandes softwares de infra-estrutura são código aberto e de graça. E paixão não vem com uma etiqueta de preço.

Então faça o que puder com o dinheiro que tem em mãos. Pense muito e determine o que é realmente essencial e o que pode viver sem. O que pode fazer com três pessoas em vez de dez? O que pode fazer com R$ 40 mil em vez de R$ 200 mil? O que pode fazer em três meses em vez de seis? O que pode fazer se puder manter seu emprego e construir sua aplicação nas horas vagas?

Restrições forçam a criatividade

Dirija com recursos limitados e será forçado a contar com restrições mais cedo e mais intensamente. E isso é uma coisa boa. Restrições dirigem inovação.

Um retorno rápido é bem improvável. Então foque em construir uma ferramenta de qualidade que você e seus clientes poderão viver com por um bom tempo.

Nunca jogue mais tempo ou dinheiro em um problema, apenas diminue o escopo.

'O mais tarde é eterno, o agora está voando.'

Lançar alguma coisa grande que está um pouco menor em escopo do que o planejado é melhor do que lançar alguma coisa medíocre e cheio de buracos porque precisou atingir uma janela mágica de prazo, orçamento e escopo.

Agora, com tudo isso dito, também é importante não ficar muito obcecado com a concorrência. Analise demais outros produtos e você vai começar a limitar sua maneira de pensar. Dê uma olhada e vá em frente para sua própria visão e suas próprias idéias. Se sua aplicação não o excita, algo está errado. Se está trabalhando nela apenas para ganhar dinheiro, isso vai aparecer. Da mesma forma, se você se sentir apaixonado pela aplicação, também vai aparecer no produto final. As pessoas conseguem ler nas entrelinhas.

Entusiasmo se manifesta prontamente, claro, mas indiferença é igualmente inesquecível. Se seu compromisso não vem com paixão genuína para o trabalho às mãos, isso se torna um vazio que é quase impossível de conciliar, não importa o quão elaborado ou atrativo é o design. —Khoi Vinh, Subtraction.com

Quanto mais enxuto for, mais fácil é para mudar

Quanto mais massa tiver um objeto, mais energia é necessária para mudar sua direção. É uma verdade tanto para o mundo dos negócios como para o mundo físico.

'Deixe as limitações lhe guiar para soluções criativas'

Nunca há suficiente para dar a volta. Sem tempo suficiente. Sem dinheiro suficiente. Sem pessoal suficiente.

Isso é uma coisa boa.

Em vez de se desesperar com essas restrições, aceite-as. Deixe que elas o guiem. Restrições incentivam inovação e forçam o foco. Em vez de tentar removê-las, use-as em seu benefício.

Diferencie-se das companhias maiores sendo amigável e pessoal

Muitas pequenas empresas cometem o erro de tentarem atuar grande. É como se elas entendessem seu tamanho como uma fraqueza que precisa ser encoberta. Muito ruim. Ser pequeno pode realmente ser uma grande vantagem, especialmente quando isto representa comunicação.

Pequenas empresas gostam de menos formalidades, menos burocracia e mais liberdade. Menores empresas são mais próximas dos clientes por padrão. Isto significa que elas podem se comunicar com seus clientes de forma mais direta e pessoal. Se a empresa é pequena, pode-se usar uma linguagem familiar ao invés de jargão. Seu site e seu produto podem ter uma voz humana ao invés de soar como um zumbido corporativo. Ser pequeno significa poder falar com os clientes, e não “se submeter a eles.”

Sempre disponível

Não importa em qual negócio você está, um bom serviço ao cliente tornou-se o maior requisito que qualquer cliente estabelecerá. Nós demandamos isso dos serviços que usamos então por que com nossos clientes seria diferente? Desde o começo nós deixamos fácil e transparente para nossos clientes contatar-nos por toda e qualquer questão que tiverem. Em nosso website nós listamos um grande número de ferramentas gratuitas que redireciona para nossos celulares e nossos cartões de visita listam os números de cada um de nós. Nós enfatizamos para nossos consumidores que eles podem nos contatar a qualquer hora independente do problema. Nossos clientes apreciam esse nível de confiança ninguém jamais abusou deste serviço. —Edward Knittel, Diretor de Vendas e Marketing, KennelSource

Faça um Mantra

Organizações precisam de pontos-guia. Precisam de linhas gerais; funcionários precisam saber a cada dia quando acordam porque estão indo trabalhar. Essas linhas devem ser curtas e doces, e bem compreensivas: Por que você existe? O que o motiva? Chamo isso de mantra – uma descrição de três ou quatro palavras de porque você existe. —Guy Kawasaki, autor (de Make Mantra)

Sucesso e satisfação estão nos detalhes

Entretanto, o sucesso não é a única coisa que encontrará nos detalhes. Também encontrará estagnação, desacordo, reuniões e atrasos. Essas coisas podem acabar com a moral e diminuir suas chances de sucesso.

Quantas vezes se encontrou travado em um único design ou elemento de código por um dia inteiro? Quantas vezes se deu conta de que o progresso que fez hoje não foi progresso real? Isso acontece quando você foca nos detalhes cedo demais no processo. Há tempo suficiente para ser um perfeccionista. Apenas faça isso mais tarde.

Não se preocupe com o tamanho da fonte do cabeçalho na primeira semana. Você não precisa empregar o tom perfeito de verde na segunda semana. Não precisa mover em três pixels o botão de "submeter" na terceira semana. Apenas coloque as coisas na página por enquanto. Então use. Garanta que funciona. Mais tarde você pode ajustar e aperfeiçoar.

Os detalhes se revelam ao se usar o que está construindo. Você verá o que precisa de mais atenção. Sentirá o que está faltando. Saberá quais crateras pavimentar porque ficará sempre caindo nelas. É quando precisa prestar atenção, e não antes.

O Diabo está nos Detalhes

Quase me cansei da atitude "entre nos detalhes imediatamente" depois de tomar algumas aulas de desenho … Se começar a desenhar os detalhes imediatamente pode ter certeza que o desenho será uma droga. De fato, você está perdendo completamente o ponto.

Você deve começar pegando as proporções corretas da cena toda. Então rascunha os grandes objetos na sua cena, indo até os menores. O rascunho deve ser bem vago nesse ponto. Então pode proceder sombreando, o que consiste em dar volume à vida. Você começa com apenas três tons (claro, médio, escuro). Isso dá um rascunho de tons. Então, para cada porção do seu desenho reavalia três tons e os aplica. Faça isso até os volumes aparecerem (requer múltiplas iterações) ...

Funciona do grande para o pequeno. Sempre. — Patrick Lafleur, Creation Object Inc. (de Signal vs. Noise)

Faça Software que tem Opinião

Seu aplicativo deve tomar partido

Algumas pessoas defendem que o software deve ser agnóstico. Dizem que é arrogante da parte dos desenvolvedores limitar a funcionalidade ou ignorar pedidos de novos recursos. Dizem que o software deve ser sempre o mais flexível possível.

Para nós isso é papo-furado. O melhor software traz consigo uma visão. O melhor software toma partido. Quando alguém usa um software, não está procurando apenas recursos, está procurando uma abordagem. Está procurando uma visão. Decida qual é sua visão e atenha-se a ela.

E lembre, se não gostarem da sua visão há um monte de outras visões por aí. Não corra atrás de quem você nunca irá contentar.

Um ótimo exemplo é o projeto original do wiki. Ward Cunningham e seus amigos deliberadamente desproveram o wiki de muitos recursos que no passado eram considerados parte indispensável da colaboração de documentos. Em vez de atribuir cada mudança do documento a uma pessoa determinada, eles removeram muito da representação visual de propriedade. Eles tornaram o conteúdo atemporal e destituído de ego. Eles decidiram que não importava quem escreveu o conteúdo ou quando ele foi escrito. E isso fez toda a diferença. Essa decisão despertou nas pessoas um senso de comunidade e foi peça-chave no sucesso da Wikipédia.

Nossos aplicativos trilharam um caminho parecido. Eles não tentam ser todas as coisas para todas as pessoas. Eles têm uma atitude. Eles vão atrás de clientes que são no fundo parceiros. Eles têm apelo para as pessoas que partilham de nossa visão. Ou se está do lado de dentro ou se está do lado de fora.

'Comece com Não'

Cada vez que você diz sim para uma funcionalidade, você está adotando um filho. Você tem que levar seu bebê através de toda uma cadeia de eventos (exemplo: design, implementação, testes etc.). Uma vez que está funcionalidade está lá, você está preso a ela. Apenas tente removê-la e veja o quão irados ficarão os clientes. Não concorde com tudo

Faça com que cada funcionalidade dê duro para ser implementada. Ponha cada uma delas à prova e mostre que é uma sobrevivente. É como no filme “O Clube da Luta”. Você deveria considerar apenas funcionalidades que estejam dispostas a ficar aguardando na porta por três dias para serem aceitas.

É por isso que você tem que começar com um não. Cada novo pedido de funcionalidade que vem até nós – ou de nós – encontra um não. Nós ouvimos mas não agimos. A resposta inicial é “agora não”. Se o pedido continua a aparecer, então sabemos que é hora de um olhar mais profundo. Somente então nós começamos a pensar na funcionalidade de fato.

E o que dizer às pessoas que reclamam quando nós não adotamos a sua idéia? Lembre-os do porque eles gostam da aplicação em primeiro lugar. “Você gosta dele porque nós dizemos não. Você gosta dele porque ele não faz outras 100 coisas. Você gosta dele porque ele não tenta agradar a todos sempre.”

Crie algo que você possa gerenciar

Deixe os clientes informarem o que é importante

Os clientes querem absolutamente tudo. Eles virão com uma avalanche de pedidos de funcionalidades. Dê uma olhada nos fóruns de nossos produtos; A categoria 'pedido de funcionalidade' sempre sobrepuja as com larga vantagem.

Nós vamos ouvir sobre "essa pequena funcionalidade extra" ou "não pode ser difícil" ou "não seria fácil colocar isso" ou "vai levar apenas uns segundos para inserí-la" ou "se você adicionar isso, eu pagaria o dobro" e assim por diante.

Claro que não podemos culpar as pessoas por pedir funcionalidades. Nós as encorajamos e queremos ouvir o que elas tem a dizer. A maior parte das funcionalidades que inserimos em nossos produtos começaram como sugestões de nossos clientes. Mas, como dissemos antes, sua primeira resposta deve ser um não. Então o que você faz com todos esses pedidos? Onde você os guarda? Como você os gerencia? Você não faz isso. Você apenas os lê e então os joga fora.

Sim, leia, jogue fora e esqueça-os. Pode soar como heresia mas os realmente importantes irão, com certeza, reaparecer. Esses são os únicos que você precisa se lembrar. Esses são os realmente esseciais. Não se preocupe em organizar e guardar cada pedido que aparecer. Deixe seus clientes serem sua memória. Se a funcionalidade for realmente necessária, eles te lembrarão até que você não consiga esquecer.

Da Idéia à Implementação

Vá do brainstorm à esboços à HTML à codificação

Aqui vai o processo que usamos para Cair na Real:

Brainstorm

Traga idéias à tona. O que este produto irá fazer? Para o Basecamp, nós olhamos para nossas próprias necessidades. Queríamos publicar atualizações de projeto. Queríamos participação dos clientes. Sabíamos que projetos tinham datas-chave. Queríamos centralizar arquivos para que as pessoas pudessem revisar coisas antigas com facilidade. Queríamos ter uma visão da figura maior, uma vista aérea do que estava acontecendo com todos os nossos projetos. Juntas, estas premissas e algumas outras, serviram como nossa fundação.

Esse estágio nao é sobre os mínimos detalhes. É sobre grandes questões. O que a aplicação precisa fazer? Como saberemos quando será útil? O que exatamente faremos? Isso é sobre idéias de alto nível, nao discussões no nível dos pixels. Nesse estágio, esses tipos de detalhe simplesmente não têm sentido. Papel de Padeiro

Esboços são rápidos, sujos e baratos e é exatamente como você quer começar. Desenhe coisas. Rabisque coisas. Caixas, círculos, linhas. Arranque as idéias da cabeça para o papel. O objetivo nesse ponto deve ser converter conceitos em designs grosseiros de interface. Esse passo é apenas sobre experimentação. Não há respostas erradas. Crie telas HTML

Faça uma versão HTML dessa funcionalidade (ou seção, ou fluxo, se for mais apropriado). Pegue algo real e publique para que todos possam ver como fica na tela.

Para o Basecamp, primeiro fizemos a tela de "postar mensagens", então a tela de "editar mensagens" e a coisa prosseguiu daí.

Não escreva nenhum código de programação ainda. Apenas faça um protótipo em html e css. A implementação vem depois.

Codifique

Quando o protótipo parecer bom e demonstrar o suficiente das funcionalidades necessárias, vá em frente e conecte o código de programação.

Durante todo esse processo, se lembre de permanecer flexível e esperar múltiplas iterações. Você deve se sentir livre para jogar fora qualquer parte entregável de qualquer passo particular e começar novamente se ela se mostrar lixo. É natural passar por esse ciclo múltiplas vezes.

Teste sua aplicação com uso do mundo real

Não tenha reuniões

Você precisa mesmo de reuniões? Reuniões geralmente acontecem quando um conceito não está claro o suficiente. Ao invés de recorrer a uma reunião, tente simplificar o conceito, para que você possa discutí-lo rapidamente por email ou IM ou Campfire. O objetivo é evitar reuniões. Cada minuto que você gasta em uma reunião é um minuto que você poderia estar trabalhando.

Não existe nada mais tóxico à produtividade do que uma reunião. Aqui vão alguns motivos:

- Elas quebram seu trabalho diário em pequenos períodos, que acabam por quebrar o fluxo do trabalho

- Elas geralmente tratam apenas de palavras e conceitos abstratos, não de coisas reais (como um trecho de código ou algum detalhe do design de interface)

- Elas geralmente tratam de uma pequena quantidade de informações por minuto

- Elas quase sempre tem uma pessoa que inevitavelmente vai fazer com que todos percam o tempo com assuntos não relacionados

- O assunto principal vai embora muito facilmente

- Freqüentemente tem pautas tão vagas que ninguém tem certeza do assunto principal

- Requerem uma preparação prévia, que quase ninguém faz

Em casos em que reuniões são realmente necessárias (faça disso um raro evento), siga estas regras simples:

```
   * Coloque um alarme pra 30 minutos. Assim que ele tocar, a reunião acabou.
Ponto final.
   * Chame o menor número de pessoas possível.
   * Nunca tenha uma reunião sem uma pauta bem clara.
```

Trabalhe com possíveis funcionários na base do "teste antes"

Uma coisa é olhar o portfólio, curriculum, exemplo de código ou trabalhos anteriores. Outra coisa é efetivamente trabalhar com alguém. Sempre que possível, faça um “test-drive” com possíveis novos membros da equipe.

Isso significa que você pode julgar pessoas pelas ações ao invés de apenas palavras. Você pode tomar decisões com base no que realmente importa:

- Qualidade do trabalho

Muitos programadores falam bonito, mas afinam na hora do “vamos ver”. Com open source, você consegue ver com detalhes as práticas e conhecimentos de programação de uma pessoa.

- Perspectiva cultural

Programar é tomar decisões. Muitas delas. Decisões são tomadas com base na cultura, nos valores e em ideais. Veja as decisões específicas feitas por um candidato enquanto está programando e testando, e veja seus argumentos na comunidade para ver se o candidato está dentro do que a empresa espera. Se não se encaixa na empresa, as decisões podem parecer erradas.

- Nivel de paixão

Por definição, envolvimento em projetos open source requerem um nível mínimo de paixão. Se não, porque outro motivo a pessoa perderia tempo na frente de um monitor? O tamanho do envolvimento em movimentos open source mostra quanto um candidato realmente se importa com programação.

- Porcentagem de finalização

Toda a inteligência, toda a cultura e paixão não se transformam em software de valor se o candidato não consegue terminá-lo. Infelizmente, muitos programadores não terminam seus projetos. Então, procure a exceção. Contrate aquele que consegue sair pela porta e está disposto a fazer as trocas pragmáticas que o trabalho exige.

- Lado social

Trabalhar com alguém por um bom período de tempo, durante tanto as horas de stress e descontração e altos e baixos vão mostrar a verdadeira personalidade do candidato. Se alguém não tem modos ou um lado sociável, deixe-os de lado.

Procure por generalistas que aprendem rápido em vez dos especialistas limitados

Nunca contrataremos alguém que seja um arquiteto de informação. É simplesmente específico demais. Com uma equipe pequena como a nossa, não faz sentido contratar pessoas com um conjunto de conhecimento tão limitado.

Equipes pequenas precisam de pessoas que possam vestir diferentes chapéis. Precisamos de designers que saibam escrever. Precisamos de programadores que entendam de design. Todos devem ter noção de como arquitetar informação (seja lá o que isso signifique). Todos precisam ter mentes organizadas. Todos precisam saber se comunicar com clientes.

E todos precisar querer e serem capazes de diminuir a marcha pela estrada. Tenha em mente que equipes pequenas eventualmente precisam mudar de direção rapidamente. Queremos alguém que possa se ajustar, aprender e fluir ao contrário de um pé-na-lama que só consegue fazer uma coisa.

Contrate bons escritores

Se está tentando decidir entre poucas pessoas para preencher uma posição, sempre contrate o melhor escritor. Não importa se essa pessoa é um designer, programador, marketing, vendedor ou o que for, essa habilidade leva a escrever mais efetivamente e concisamente código, design, emails, mensagens instantâneas e mais.

Isso porque ser um bom escritor é mais do que apenas palavras. Bons escritores sabem como se comunicar. Eles tornam as coisas mais fáceis de entender. Eles podem se colocar no lugar dos outros. Eles sabem o que omitir. Eles pensam claramente. E essas são as qualidades que você precisa.

Uma Mente Organizada

Boas habilidades de escrita são um indicador de uma mente organizada que é capaz de arranjar informação e argumentos de uma maneira sistemática e também ajudar (não fazer) outras pessoas a entender as coisas. Isso aparece no código, comunicação pessoal, mensagens instantâneas (para aqueles colaboradores de longa distância) e até esses conceitos exotéricos como profissionalismo e confiança.

—Dustin J. Mitchell, developer (de Signal vs. Noise) Escrita Clara leva a Pensamento

Escrita clara leva a pensamento claro. Você não sabe o que sabe até tentar expressar esse conhecimento. Boa escrita é em parte uma questão de caráter. Em vez de fazer o que é fácil para você, faça o que é mais fácil para seu leitor.

—Michael A. Covington, professor de ciências da computação da Universidade da Geórgia (de Como Escrever mais Claramente, Pensar mais Claramente e aprender Material Complexo mais Facilmente)

Faça Design para quando as coisas derem errado

Vamos admitir: As coisa vão dar errado online. Não importa o quão cuidadoso você faça o design de sua aplicação, não importa quanto teste fizer, os clientes ainda vão encontrar problemas. Então como você gerencia essas quedas inevitáveis? Com design defensivo.

Escolha ferramentas que estimulem e motive o seu time

Um programador feliz é um programador produtivo. É por isso que nós otimizamos para felicidade e você deveria fazer o mesmo. Não escolha as ferramentas e práticas baseado simplesmente no padrão do mercado ou métricas de desempenho. Avalie os atributos intangiveis: a ferramenta foi criada com paixão, orgulho e dedicação?. Você seria feliz trabalhando neste ambiente oito horas por dia?

O Código Fala

Ouça quando seu código diz "não"

Ouça seu código. Ele oferecerá sugestões. Ele irá dizer "não". Ele lhe dirá onde ficam as armadilhas. Ele irá sugerir novas maneiras de fazer as coisas. Ele irá ajudá-lo a se manter em um modelo de menos software.

Uma nova funcionalidade está requerendo semanas de tempo e milhares de linhas de código? Isso é

seu código lhe dizendo que provavemente existe uma maneira melhor. Existe uma maneira simples de codificar alguma coisa em uma hora em vez de uma maneira complicada que consumirá dez horas? Novamente, esse é seu código o guiando. Ouça.

Seu código pode guiá-lo a consertos que são baratos e leves. Preste atenção quando um caminho mais fácil emerge. Claro, a funcionalidade que é fácil de fazer pode não ser exatamente a mesma que você originalmente tinha em mente, mas e daí? Se funciona bem o suficiente e lhe dá mais tempo para trabalhar em outra coisa, é um ganhador.

Ouça

Não se preocupe com o design, se ouvir seu código um bom design vai aparecer ... Ouça as pessoas técnicas. Se eles estão reclamando sobre a dificuldade de fazer mudanças, então leve essas reclamações a sério e lhes dê tempo para consertar as coisas.

—Martin Fowler, Cientista Chefe, ThoughtWorks (de Is Design Dead?)

Abra as Portas

Publique dados para o mundo via RSS, APIs, etc.

Não tente prender seus usuários. Deixe que eles possam ter acesso a suas informações quando quiserem, da forma que preferirem. Para tal, você precisa deixar de lado a idéia de manter os dados de seus usuários trancados a sete chaves. Em vez disso, deixe que a informação flua. Garanta o acesso à informação através de feeds RSS. Ofereça APIs que permitam a terceiros construir aplicações integradas à sua. Tais atitudes tornarão a vida dos usuários mais conveniente e expandirão as possibilidades do que sua aplicação é capaz de fazer.

No passado, as pessoas acostumaram-se a pensar nos feeds RSS apenas como uma boa maneira de se agregar conteúdo de sites de blogs e sites de notícia. Contudo, os feeds são mais poderosos que isto. Eles também podem permitir ao usuário manter-se atualizado sobre mudanças internas à aplicação sem a necessidade de logar-se repetidas vezes. Através do site do Basecamp, por exemplo, o usuário pode cadastrar sua url em um agregador de RSS e assim receber notificações de mensagens de projetos, listas de tarefas e objetivos sem a necessidade de conectar-se constantemente ao site em busca de informações atualizadas.

APIs permitem que desenvolvedores construam plugins adicionais à sua aplicação, que geralmente agregam valor ao seu produto. Por exemplo, a API disponibilizada pelo Backpack foi utilizada pela Chipt Productions na construção de um widget para o Mac os X. A pequena aplicação permite aos usuários adicionar e editar lembretes, listagens de items e muito mais a partir de seus desktops. Muitos usuários apontaram o widget como uma ótima ferramenta, e alguns mesmo apontaram-no como um fator decisivo na escolha da utilização do Backpack.

Outros bons exemplos de empresas que liberaram dados como uma maneira de conseguir um 'efeito bumerangue':

- A API do Google Maps permitiu o surgimento de toda sorte de pequenas aplicações que recuperam dados de outras fontes (ex.: uma listagem de apartamentos) e os exibem em um mapa.

- Linkrolls oferece aos usuários exibir seus últimos bookmarks do del.icio.us em seu próprio site.

- O Flickr permite que outros negócios acessem as suas APIs comerciais, de forma a permitir aos usuários comprar livros de fotos, posters, backups em DVD e selos. "O objetivo é manter as portas completamente abertas e permitir o maior número possível de possibilidades de utilização de suas fotos", diz Stewart Butterfield, do Flickr.

Um Widget Faz a Diferença

Quando a 37signals lançou o Backpack, há algum tempo atrás, minha primeira impressão foi… er... bem...

Ocorreu mais ou menos na época em que a Chipt Productions lançava um widget Backpack para o Sistema Operacional Tiger — que parecia interessante demais para passar despercebido — com isso dei uma segunda olhada no Backpack. O resultado? Uma grande diferença.

Hoje, sempre que uma nova idéia surge, abro o widget, digito e salvo — e pronto. Recebo algum e-mail com algo que devo fazer? Abro o widget, digito e salvo — e pronto. O widget tornou-se um tipo de bloco de notas indispensável, que instalo em todo Mac que uso. E por se tratar de uma aplicação totalmente web, não há necessidade de nenhum tipo de controle de versão ou sincronizaçao de dados — apenas a fluidez de digitar-se dados sem ter que se preocupar em saber para onde os dados foram, nem como acessá-los mais tarde. —Todd Dominey, fundador, Dominey Design (de Trying on Backpack)

Amostra Grátis

Dê alguma coisa de graça

É um mundo barulhento lá fora. Para que as pessoas o notem no meio da multidão, dê alguma coisa de graça.

Empresas espertas sabem que dar brindes é uma excelente maneira de fisgar clientes. Veja a Apple. Eles oferecem o software iTunes de graça de forma a gerar demanda para o iPod e a loja de música iTunes. No mundo offline, as lojas fazem a mesma coisa. A Starbucks diz que uma nova compra é estimulada para cada cinco amostras de bebidas que eles dão aos clientes. Nada mau.

Para nós, Writeboard e Ta-da list são aplicativos completamente grátis que usamos para colocar as pessoas no caminho para usar nossos outros produtos. Adicionalmente, sempre oferecemos algum tipo de versão grátis de todos os nossos aplicativos.

Queremos que as pessoas experimentem o produto, a interface, a utilidade do que construímos. Uma vez fisgados, eles são muito mais propensos a atualizar para um dos planos pagos (que permitem mais projetos ou páginas e dá acesso a funcionalidades adicionais como upload de arquivos e encriptação de dados com SSL). Pedacinhos

Faça pedacinhos: crie ofertas especializadas, pequenas para que os clientes mordam. Subdivida pelo menos um produto ou serviço em pedacinhos que são baratos, fáceis ou divertidos.

—Ben McConnell e Jackie Huba, autores do Church of the Customer Blog (de What is customer evangelism?)

Dê Sua Música de Maior Sucesso

Considere doar uma de suas músicas (por álbum) como download gratuito promocional para o mundo – para ser como um trailer de cinema – como o single de sucesso enviado ao rádio – a música que faz as pessoas quererem comprar sua música.

Não se preocupe com pirataria dessa música. Deixe as pessoas tocarem, copiarem, compartilharem. Tenha a confiança que, se o mundo a ouviu, irão pagar por mais. —Derek Sivers, presidente e programador, CD Baby e HostBaby (de Free Promo Track)

Um Poderoso Site Promocional

Vá do Trailer para a Prévia para o Lançamento

A melhor ferramenta promocional é um grande produto. A palavra vai se espalhar se tivermos uma aplicação que as pessoas acham realmente útil.

Ainda assim, precisamos de um bom site promocional também. O que devemos incluir nesse site? Algumas idéias:

```
   * Apresentação: Explique sobre a aplicação e seus benefícios.
   * Turismo: Guie as pessoas pelas várias funcionalidades
   * Fotos de tela e vídeos: Mostre às pessoas como sua aplicação realmente se
parece e como usá-la.
   * Manifesto: Explique a filosofia e idéias por trás dela.
   * Estudos de Caso: Dê exemplos reais que mostram o que é possível.
   * Euforia: Frases testimoniais de clientes, revisões, imprensa, etc.
   * Fórum: Ofereça um local para membros da comunidades se ajudarem uns aos
outros.
   * Precificação e Assinatura": Leve as pessoas à aplicação o mais rápido
possível.
   * Weblog": Blogs mantém seu site atualizado com notícias, dicas, etc.
```

Sinta a Dor

Derrube as paredes entre suporte e desenvolvimento

No negócio de restaurantes, existe uma enorme diferença entre aqueles que trabalham na cozinha daqueles que estão na linha de frente lidando com clientes. É importante para ambos os lados entender e simpatizar com o outro. É por isso que escolas de culinária e restaurantes normalmente terão chefs trabalhando como garçons para que a equipe da cozinha possa interagir com clientes e ver como é realmente estar na linha de frente.

Muitas empresas desenvolvedoras de software tem uma divisão similar. Designers e programadores trabalham na "cozinha" enquanto o suporte lida com clientes. Infelizmente, isso significa que chefs de software nunca ouvem o que o cliente realmente está dizendo. Isso é problemático porque ouvir clientes é a melhor maneira de se ligar nas partes fortes e fracas do seu produto.

A solução? Evite construir paredes entre seus clientes e a equipe de desenvolvimento/design. Não terceirize o suporte a seus clientes. Faça você mesmo o suporte. Você e sua equipe inteira, devem saber o que seu cliente está dizendo. Quando seu cliente está incomodado, você precisa saber disso. Você pecisa ouvir as reclamações. Você precisa ficar incomodado também.

Na 37signals, todos os e-mails de suporte são respondidos pessoalmente pelo pessoal que realmente construiu o produto. Por que? Primeiro, isso fornece melhor suporte aos clientes. Eles estão recebendo uma resposta diretamente do cérebro de alguém que construiu a aplicação. Além disso, isso nos mantém em contato com a pessoa que usa nossos produtos e com os problemas que estão encontrando. Quando estão frustrados, nós ficamos frustrados. Podemos dizer sinceramente que "eu sinto sua dor".

Pode ser tentador se apoiar em análises estatísticas para revelar seus pontos problemáticos. Mas estatísticas não são como vozes reais. Você precisa eliminar a maior quantidade possível de atravessadores entre você e as vozes reais de seus clientes.

As linhas de frente são onde a ação está. Vá até lá. Faça seus chefs trabalharem como garçons. Leia e-mails de clientes, ouça suas frustrações, escute suas sugestões e aprenda com elas.

Treinamento Zero

Use ajuda em contexto e FAQs para que seu produto não precise de um manual ou treinamento

Você não precisa de um manual para usar o Yahoo! ou Google ou Amazon. Então por que você não pode construir um produto que não requer manual? Se esforce para construir uma ferramenta que requer treinamento zero. Como fazer isso? Bem, como mencionamos antes, você começa mantendo tudo simples. Quanto menos complexa for sua aplicação, menos precisará ajudar as pessoas sem necessidade. Depois disso, uma grande maneira de suporte pró-ativo é usando ajuda em contexto e FAQs em potenciais pontos de confusão.

Por exemplo, oferecemos suporte pró-ativo na tela que permite as pessoas a fazer upload de seus logotipos ao Basecamp. Algumas pessoas experimentaram um problema onde continuavam vendo um logotipo antigo por causa do cache do browser. Então, próxima à área de "envie seu logotipo", adicionamos um link a um FAQ que instruía os clientes a forçar um recarregamento de seus browsers para ver o novo logotipo. Antes de fazermos isso recebíamos 5 e-mails por dia sobre esse problema. Agora, não recebemos nenhum.

Em Fórum Afinado

Use fórums ou chats para deixar os clientes se ajudarem

Fórum e chats de grupo baseados na web são uma grande maneira de deixar clientes fazerem perguntar e ajudar uns aos outros. Eliminando o intermediário – esse é você – você fornece uma linha aberta de comunicação e economiza seu tempo no processo.

Em nossos fóruns de produtos, os clientes publicam dicas e truques, requisições de funcionalidades, histórias e mais coisas. Nós aparecemos de tempos em tempos para oferecer assistência, mas os fóruns são principalmente um lugar para a comunidade se ajudar e compartilhar experiências com o produto.

Você ficará surpreso com quantas pessoas querem se ajudar.

Original em:

http://gettingreal.37signals.com/GR_por.php

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Intelig%C3%AAnciaEmocional/Resumo_do_Livro_Caindo_na_Real"

Aplicativos em PHP/Instalação e configurações do ambiente/Via Xampp

De Wikibooks

< Aplicativos em PHP | Instalação e configurações do ambiente

Conteúdo

- 1 O que é o Xampp?
- 2 Sistemas Operacionais Suportados
- 3 Utilitários
- 4 Instalação
- 5 No Windows
- 6 No Linux
- 7 Usando o Xampp
- 8 Testando PHP
- 9 Instalar PostgreSQL e integrar com PHP
- 10 Tornando o Xampp mais Seguro
  - 10.1 Usando o Security do Xampp
  - 10.2 Adicionando senha ao MySQL e phpMyAdmin
- 11 Mudando o DocumentRoot
- 12 Adicionando Domínio Virtual no Apache com Xampp
- 13 Apache 2 comentado
  - 13.1 Suporte ao conjunto de caracteres da ISO-8859-1 direto no Apache (httpd.conf)
- 14 Suporte ao conjunto de caracteres da ISO-8859-1 direto nos scripts

O que é o Xampp?

O Xampp é um utilitário que instala o Apache, PHP (PEAR e PECL), MySQL, Perl, phpMyAdmin, proFtpd e outros. Atualmente se encontra na versão XAMPP 1.6.2 para o windows e já vem configurado com:

- MySQL 5.0.41
- PHP 5.2.2
- PHP 4.4.7
- phpMyAdmin 2.10.1
- e muito mais

Sistemas Operacionais Suportados

Tem versões para Linux, Windows e outros SOs.

Utilitários

Traz ainda em sua index (http://127.0.0.1) um menu com a relação dos seus utilitários e alguns exemplos em php e perl. Este menu é facilmente incrementado ao se editar os arquivos .inc em:

C:\Arquivos de programas\xampp\HTDOCS\xampp.

Instalação

Faça o download do site oficial de acordo com o seu sistema (http://xampp.sf.net) ou se preferir em português do Brasil: http://www.apachefriends.org/pt_br/xampp.html

No Windows

Execute o instalador

Diretório web default = C:\Arquivos de programas\xampp\HTDOCS

php.ini - C:\Arquivos de programas\xampp\apache\bin

httpd.conf - C:\Arquivos de programas\xampp\apache\conf

No Linux

Descompacte em /opt com

tar zxpvf xampp.*.tar.gz -C /opt

Diretório web - /opt/lampp/htdocs

php.ini e httpd.conf - /opt/lampp/etc

ATENÇÃO: Não esqueça de, após instalar o xampp, mudar as senhas default do apache, mysql e do próprio xampp. Ignorando este aviso, o xampp ficaria perigosamente exposto à invasões ou afins. (para mais informações de como proceder para mudar as senhas, vide a seção abaixo - Tornando o xampp mais seguro)

Usando o Xampp

No windows o executável abre um menu onde podemos executar os serviços separadamente. Após executar um ícone fica na barra de tarefas, o que facilita o gerenciamento.

No Linux – execute o arquivo "/opt/lampp/lampp serviço ação", onde:

serviço – um dos serviços disponíveis (httpd, mysql, etc)

ação – start, stop, status

/opt/lampp/lampp start (inicia todos os serviços disponíveis)

Configurando o Servidor de FTP (proFtpd)

editar /opt/lampp/etc/proftpd.conf e alterar usuário, senha e diretório a gosto

O diretório default é o /opt/lampp/htdocs

Sugestão: criar dentro do htdocs um sub com nome ftp.

A senha do proftpd não pode ser trocada manualmente, pois é criptografada.

Para trocar devemos usar o security:

/opt/lampp/lampp security

Dica: Se você não quiser ter que digitar o caminho completo (/opt/lampp/lampp start) sempre que quiser lançar o servidor xampp, você pode criar um link para o executável no "path" do seu sistema com um simples:

$ sudo ln -s /opt/lampp/lampp /usr/local/bin/lampp

Com isso, um

$ sudo lampp start

bastará para lançar o servidor.

Testando PHP

Tanto no windows quanto no linux, startar o apache e abrir o browser apontando para http://localhost ou http://127.0.0.1

Se tudo correu bem verá a index do Xampp com todos os seus recursos,onde poderá inclusive optar pelo idioma Português do Brasil.

Instalar PostgreSQL e integrar com PHP

Download do site oficial (http://www.postgresql.org), escolhendo pelo sistema operacional.

Instalação

Windows – execute o instalador

Linux – Para distribuições com instalação de binários fica mais fácil

Instalar dos fontes. Apenas descompacte e leia o arquivo INSTALL, onde encontrará passos resumidos para a instalação.

Configuração

Para configurar que máquinas terão acesso ao postgresql edite o arquivo pg_hba.conf, que fica no subdiretório data do postgresql.

Configure acesso via TCP/IP no arquivo postgresql.conf que fica no mesmo diretório do pg_hba.conf (em algumas distribuições fica no /etc/postgresql).

Integrando com PHP

No Windows descomente a linha com “extension=php_pgsql.dll” e restarte o apache. No Linux o suporte já vem habilitado por default.

Tornando o Xampp mais Seguro

Usando o Security do Xampp

No Linux - Abra a interface web (127.0.0.1)

- Clique no link Segurança à esquerda e leia as instruções

- Para ativar a segurança execute como super usuário:

/opt/lampp/lampp security

E ative tudo que desejar.

Após isso, para acessar o phpMyAdmin de forma mais confortável, edite o arquivo:

/opt/lampp/lib/xampp.users e alterar o nome do user default de "lamp" para o seu.

Então ao acessar entre com seu usuário e senha (a que cadastrou ao adicionar a segurança no Xampp para acessar o sistema de arquivos) e logo em seguida entre com root e senha (do banco).

Adicionando senha ao MySQL e phpMyAdmin

Caso pretendamos apenas adicionar senha ao MySQL e deixar o Xampp e PHP como estão...

O MYSQL inicia sem senha para o usuário "root". No PHP você pode conectar ao MYSQL através do comando: mysql_connect("localhost","root",""); Se você quiser atribuir uma senha para o usuário "root" no MYSQL, por favor, use o comando "mysqladmin" no console. Por exemplo:

\...\xampp\mysql\bin\mysqladmin -u root password secret

Atenção: Depois de alterar a senha do usuário "root", não esqueça de configurar o PHPMyAdmin. Procure o arquivo "config.inc.php" em \...\xampp\phpmyadmin\ e edite as seguintes linhas:

$cfg['Servers'][$i]['user'] = 'root'; // MySQL SuperUser $cfg['Servers'][$i]['auth_type'] = 'http'; // HTTP MySQL authentification

Agora a senha correta para o usuário "root" é requisitada, antes do PHPMyAdmin iniciar.

Obs.: Apenas copiei e colei da ótima FAQ do Xampp em pt-br (http://www.apachefriends.org/pt_br/xampp-windows.html).

Mudando o DocumentRoot

Em algumas vezes é mais cômodo ou então mais seguro alterar o diretório default do DocumentRoot do Apache. Neste caso, **para o Linux** proceder da seguinte forma:

Mude o caminho para /home/ribafs:

- Acesse o terminal

- sudo gedit /opt/lampp/etc/httpd.conf

- Tecle Ctrl+F para localizar "htdocs"

- Troque as duas ocorrências com htdocs para "/home/ribafs/htdocs"

- Crie a pasta: md /home/ribafs/htdocs

- Então no terminal novamente execute:

sudo cp -Ra /opt/lampp/htdocs /home/ribafs/htdocs

- Mude as permissões para o usuário "ribafs" assim:

sudo chown -R ribafs:ribafs /home/ribafs/htdocs

sudo chmod -R 705 /home/ribafs/htdocs

Então

cd /home/ribafs/htdocs/xampp

chmod 777 lang.tmp

Adicionando Domínio Virtual no Apache com Xampp

Mesmo estando usando o PHP localmente podemos ter um outro domínio ou subdomínio. Quando instalamos o PHP via Xampp localmente, nós temos o IP 127.0.0.1 ou o nome localhost, de forma que abrimos no navegador com http://127.0.0.1 ou http://localhost. Podemos também ter localmente um site que abre em outro endereço como, por exemplo, http://ribafs.localhost. Como realizar isso?

Os passos seguintes são um resumo do turotial:

Preparando o terreno para o desenvolvimento em php com o Xampp no Ubuntu

- Edite o seu /etc/hosts e adicione a linha:

-- 127.0.0.1 ribafs.localhost

- Edite o /opt/lampp/etc/httpd.conf e descomente a linha:

-- Include etc/extra/httpd-vhosts.conf

- Edite o /opt/lampp/etc/extra/httpd-vhosts.conf e insira o trecho de código:

```
<VirtualHost *:80>
DocumentRoot /home/ribafs/htdocs/ribafssite
ServerName ribafs.localhost
ErrorLog logs/ribafs.localhost-error_log
CustomLog logs/ribafs.localhost-access_log common
</VirtualHost>
```

Lembre de preservar o trecho do localhost:

```
<VirtualHost *:80>
ServerAdmin ribafs@yahoo.com
DocumentRoot /home/ribafs/htdocs
ServerName localhost
ErrorLog logs/localhost-error_log
CustomLog logs/localhost-access_log common
</VirtualHost>
```

Então reinicie o Apache.

Agora você tem um domínio virtual chamado ribafs.localhost, que inclusive atende via ping ribafs.localhost e pode ser chamado pelo navegador:

http://ribafs.localhost

No diretório indicado em DocumentRoot /home/ribafs/htdocs/ribafssite podemos adicionar o conteúdo de outro site.

Apache 2 comentado

httpd.conf todo comentado e com muitas boas dicas. Vale a pena conferir, quer tenha instalado com Xampp ou manualmente.

http://www.vivaolinux.com.br/conf/verConf.php?codigo=426

Suporte ao conjunto de caracteres da ISO-8859-1 direto no Apache (httpd.conf)

Gostaria de destacar o suporte ao nosso idioma como default nas linhas:

Basta adicionar estas linhas ao final do httpd.conf e reiniciar o apache. Ressalte-se que abaixo tem bem mais que apenas o suporte ao nosso conjunto de caracteres.

```
# Adicione essa linha para não ter problemas com acentuação
# mordecai
AddDefaultCharset ISO-8859-1

AddCharset ISO-8859-1  .iso8859-1  .latin1
AddCharset ISO-8859-2  .iso8859-2  .latin2 .cen
AddCharset ISO-8859-3  .iso8859-3  .latin3
AddCharset ISO-8859-4  .iso8859-4  .latin4
AddCharset ISO-8859-5  .iso8859-5  .latin5 .cyr .iso-ru
AddCharset ISO-8859-6  .iso8859-6  .latin6 .arb
AddCharset ISO-8859-7  .iso8859-7  .latin7 .grk
AddCharset ISO-8859-8  .iso8859-8  .latin8 .heb
AddCharset ISO-8859-9  .iso8859-9  .latin9 .trk
```

```
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5        .Big5       .big5
AddCharset WINDOWS-1251 .cp-1251   .win-1251
AddCharset CP866       .cp866
AddCharset KOI8-r      .koi8-r .koi8-ru
AddCharset KOI8-ru     .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8       .utf8

AddCharset GB2312      .gb2312 .gb
AddCharset utf-7       .utf7
AddCharset utf-8       .utf8
AddCharset big5        .big5 .b5
AddCharset EUC-TW      .euc-tw
AddCharset EUC-JP      .euc-jp
AddCharset EUC-KR      .euc-kr
AddCharset shift_jis   .sjis

AddType application/x-tar .tgz
AddType application/x-httpd-php .php

AddHandler imap-file map

AddHandler type-map var

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-
carefully
BrowserMatch "^WebDrive" redirect-carefully
```

Suporte ao conjunto de caracteres da ISO-8859-1 direto nos scripts

Caso não tenha acesso ao servidor ou queira dar suporte apenas a determinados scripts, utilize uma das linhas abaixo, dependendo da sua linguagem):

Dica encontrada em: http://forum.imasters.com.br/index.php?showtopic=173603

PHP - <?php header("Content-Type: text/html; charset=ISO-8859-1",true) ?>

ASP - <% Response.Charset="ISO-8859-1" %>

JSP - <%@ page contentType="text/html; charset=ISO-8859-1" %>

HTML/XML - <?xml version="1.0" encoding="UTF-8"?>

Aplicativos em PHP/Instalação e configurações do ambiente/Individualmente

De Wikibooks

< Aplicativos em PHP | Instalação e configurações do ambiente

Conteúdo

- 1 Instalando o PHP dos Fontes ou pegando os pacotes individualmente
- 2 No Viva o Linux
- 3 Instalando PHP4 e PHP5 ao Mesmo Tempo
  - 3.1 Instalando PHP 4 e 5 em Windows
  - 3.2 Como Instalar o apache + PHP 4 e 5 + mysql e configurar
  - 3.3 Running Apache2 With PHP5 And PHP4 At The Same Time (Linux)
- 4 Instalação do Apache, PHP, MySQL e PostgreSQL no Windows
  - 4.1 1) Instalação e Configuração do Apache
  - 4.2 2) Instalação e Configuração PHP
  - 4.3 3) Instalação e Configuração do MySQL
  - 4.4 4) Instalação e Configuração do PostgreSQL

Instalando o PHP dos Fontes ou pegando os pacotes individualmente

Aqui seguem vários links com tutoriais de como instalar o ambiente para uso do PHP pegando cada um dos pacotes, inclusive compilando para Windows e para Linux.

Também podemos utilizar diversos outros pacotes instaladores existentes como EasyPHP e outros, alertando que em sua maioria existem apenas para Windows, com algumas exceções como é o caso do

Uniform Server

Veja esta dica do Haroldo Barbosa sobre o Uniform Server:

http://www.dicas-l.com.br/dicas-l/20070519.php

Agora os tutoriais sobre a instalação do ambiente para o PHP:

Quem precisa de um controle rigoroso sobre cada pacote e sobre as características de cada um destes, para ter uma maior segurança e conhecimento dos mesmos, nestes casos a instalação de cada pacote individualmente e compilando dos fontes para controle mais rigoroso do comportamento é o caminho mais indicado.

No Viva o Linux

http://www.vivaolinux.com.br

Apache 2.0 + PHP + PostgreSQL no Slackware - http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1221

Debian com Apache, PHP4, PHP5 e MySQL - http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1366

Instalação do Apache, MySQL e PHP - http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=4617

Instalando Apache2, PHP4 e MySQL - http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1304

Instalando e configurando o Apache, PHP e MySQL no Slackware 10.1 - http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=4091

Instalando servidor Apache + PHP + MySQL + phpMyadmin + no-ip no Ubuntu 6.10 Server - http://www.vivaolinux.com.br/dicas/verDica.php?codigo=8474

PHP5 + Apache2 no Ubuntu 7.04, com suporte ao MySQL e SQL Server - http://www.vivaolinux.com.br/artigos/impressora.php?codigo=6578

Instalando o Apache + PHP + MySQL no Windows - http://www.guiadohardware.net/tutoriais/apache-php-mysql-windows/

Apache2 + PHP + PostgreSQL + phpPgAdmin - http://www.linuxajuda.org/site/node/99

Apache 2 com php5 e php4 juntos - http://www.vivaolinux.com.br/dicas/impressora.php?codigo=8801

Instalando PHP4 e PHP5 ao Mesmo Tempo

Instalando PHP 4 e 5 em Windows

http://artigos.tekever.eu/ver/?165

Como Instalar o apache + PHP 4 e 5 + mysql e configurar

http://forum.imasters.com.br/index.php?showtopic=125190

Running Apache2 With PHP5 And PHP4 At The Same Time (Linux)

http://www.howtoforge.com/apache2_with_php5_and_php4

Instalação do Apache, PHP, MySQL e PostgreSQL no Windows

(Preferentemente XP, 2K ou Server)

1) Instalação e Configuração do Apache

Fazer o download (ver link ao final) e instalar normalmente (sugestão: c:\apache).

Após instalar edite o arquivo "...\conf\httpd.conf"

Altere a linha: DocumentRoot "... /htdocs"

Para: DocumentRoot "C:/www"

Alterar: DirectoryIndex index.html

Para: DirectoryIndex index.php index.html

Procure a linha com: <Directory "C:...htdocs">

Para: <Directory "C:/www">

Adicione ao final:

```
# De acordo com as intruções do arquivo "install.txt" do PHP:
# Apache como modulo no PHP 5
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php
Action application/x-httpd-php "/php/php-cgi.exe"
LoadModule php5_module "c:/php/php5apache.dll"
# specify the directory where php.ini is
SetEnv PHPRC C:/php
```

Testando o Apache:

Abra no browser com a seguinte URL: http://localhost

Parar, Startar e Restartar

```
apache -k start
apache -k stop
apache -k restart
apache -k shutdown
```

2) Instalação e Configuração PHP

Recomendação. Baixe e descompacte a versão 5.0.3 ou superior no formato zip (por ser mais completa) no diretório c:\php (sugestão):

Faça uma cópia do arquivo "php.ini-recommended" que se encontra no raiz do php para "php.ini" e configure o arquivo php.ini alterando:

Comente esta linha: error_reporting = E_ALL

Adicionando um ; em seu início e Descomente esta:

error_reporting = E_ALL & ~E_NOTICE, Removendo o ;

Altere também a linha "display_errors = On" Para Off ao colocar o site em produção por conta da segurança.

Altere:

include_path = ".;c
\php\include"

Descomentando a linha de baixo (include_path). Para ficar assim:

include_path = ".;c:\php

Altere esta: doc_root =

Para: doc_root = c:\www

Altere esta: extension_dir = "./"

Para: extension_dir = "c:\php\ext"

Descomenta as linhas:

extension=php_mysql.dll

Para dar suporte ao mysql e ao PostgreSQL e ao PDF

Obs.: A versão php-5.0.3-Win32.zip não traz a extensão PDF.

Altere a linha: ;session.save_path = "/tmp"

Descomentando e mudando o diretório:

session.save_path = "c:\windows\temp" ou "c:\winnt\temp"

Em tempo de desenvolvimento é uma boa ajuda para os desenvolvedores:

display_errors = On

Opcionalmente:

magic_quotes_gpc = On

Copiar as DLLs "libmysql.dll" e "php5ts.dll" do diretório do PHP para:

c:\windown\system32 (ou correspondente do seu sistema).

Copiar o php.ini para o c:\windows

Para ter as variáveis globais registradas sete: register_globals = On

Terstando:

Crie um arquivo "c:\www\teste.php" com apenas a linha abaixo:

<?php phpinfo(); ?>

Abra no browser com a seguinte URL: http://localhost/teste.php

3) Instalação e Configuração do MySQL

Faça o download de www.mysql.org:

Selecione a versão estável e recomendada:

Windows Essentials (x86) 4.1.14 16.4M Pick a mirror

Windows (x86) 4.1.14 37.0M Pick a mirror

Without installer (unzip in C:\) 4.1.14 38.8M Pick a mirror

Escolha a Segunda opção para Windows

Apenas descompacte e instale, efetuando as devidas configurações, com senha eacesso a TCP/IP.

Caso tenha algum firewall rodando, tome o cuidado de habilitar o acesso ao banco ou desabilite seu firewall antes de instalar.

Após instalar e configurar o serviço MySQL já estará no ar e pronto para uso.

4) Instalação e Configuração do PostgreSQL

Baixar a versão atual.

Espelhos do Postgresql disponíveis em http://www.postgresql.org/mirrors-ftp.html.

Obs.: Caso tenha dificuldade com o Internet Explorer, já que todos os mirrors são de FTP, use o Firefox:

http://www.mozilla.org

Na Tela Service Configuration

Apenas clique em Next deixando a senha em branco que o sistema cadastra uma (esta é a senha apenas para o serviço).

Na Tela Initialise Database Cluster

Locale - Troquei para Portuguese, Brasil mas deixando C não vi diferença

Encoding - Mudei para LATIN1 (para aceitar acentos).

Superuser name - Usuário do banco de dados. Administrador do banco de dados.

```
Pode deixar o user default (como é apenas para testes e desenvolvimento!).
```

Password - Esta senha é obrigatória (4 caracteres ou mais) e entraremos com ela e o usuário acima ao entrar no banco. Sugestão para testes "postgres".

Exemplo no prompt:

```
psql -U postgres nomedobanco
```

Então clique em Next.

Enable procedural language.

Apenas next.

Enable contrib modules

Marque outros apenas se tiver conhecimento. Senão Next.

Ready to install

Next.

Configurando Prompt de Comando

```
- Abra o prompt de comando
- Clique na barra de títulos
- Clique em Propriedades
- Clique na aba Fonte
- Altere para Lucida (tamanho 12)
- Ao ser interrogado sobre as alterações marque "Modificar o atalho que
modificou esta janela" para que a alteração seja permanente.
```

Alterando a Página de Código:

Execute no prompt de comando:

cmd.exe /c chcp 1252

Como aqui a coisa não ficou permanente e não sei "ainda" como fazer eu criei um pequeno arquivo de lote apenas com a linha acima (cmd.exe /c chcp 1252) e copiei para o diretório bin do PG.

Outra boa sugestão é colocar o diretório c:\pg\bin no path para poder usar os comandos do PG mais à vontade.

Importando de Dentro do Banco

Na versão Beta apenas tive que me acostumar com a importação de dentro do banco:

\i 'c:\\diretorio\\arquivo.sql'

Acredito que será corrigido na versão final.

Obs.: Como estes passos foram executados em uma versão anterior a atual, pode ser que alguns fatores tenham evoluído.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Instala%C3%A7%C3%A3o_e_configura%C3%A7%C3%B5es_do_ambiente/Individualmente"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/Eclipse com PHPEclipse, Aptana e outros plugins

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

Conteúdo

- 1 Introdução
- 2 Características do PHPEclipse
- 3 Alterando o Help Online para Português do Brasil
- 4 Instalações
  - 4.1 Instalação do Eclipse
  - 4.2 Instalação do PHPEclipse
  - 4.3 Instalação do JSEclipse
  - 4.4 Instalação do EclipseTidy
  - 4.5 Instalação do Quantum DB
  - 4.6 Configuração do PHPEclipse no Eclipse
  - 4.7 Atualizando o Eclipse e seus plugins
  - 4.8 Criando um Novo Projeto e PHP
  - 4.9 Adicionando arquivos PHP ao Projeto criado
  - 4.10 Adicionando arquivos HTML e JavaScript
  - 4.11 Auto_completando o código
  - 4.12 Debugando Scripts PHP
  - 4.13 Configuração do PHP e Apache para melhor Debug
  - 4.14 Tipos de Erros e avisos do PHP
- 5 Using PHPEclipse : Installation : Installing the DBG Debugger

Introdução

O desenvolvimento de um aplicativo Web em PHP requer basicamente conhecimentos nas seguintes áreas: HTML, JavaScript, CSS, PHP e de um banco de dados. Com a IDE Eclipse podemos juntar alguns plugins e formar um ótimo ambiente de desenvolvimento.

Um recurso muito importante em uma ferramenta web é um bom debugger, pois com o que se perde muito tempo é com a caça aos bugs. O melhor debugger que testei foi o do PHPEclipse.

Agora comparemos o trabalho em um editor de texto comum e no editor do PHPEclipse. No editor de texto normalmente se cria um script até o final e então manda-se para o servidor para testar no browser.

No PHPEclipse a cada linha digitada temos uma ajuda do debugger (caso haja algum erro ele avisa e até corrige com uma solução de sugestão). Tem dúvida sobre alguma palavra-chave? Deixe o cursos de texto sobre a mesma e tecle Ctrl+Shift+H para receber uma página da documentação oficial do PHP, com sintaxe, exemplos, etc. Isso até no seu idioma nativo, como veremos adiante.

Auto-completa código HTML, JavaScript e PHP. Gerenciamento do banco de dados integrado (QuantumDB), Modelagem de dados (Azurry Clay) e muitas outras facilidades integradas.

Com a intenção de encontrar um ambiente de desenvolvimento que torne o trabalho com PHP mais produtivo, testei diversos editores e IDEs, inclusive alguns comerciais, como o ZendStudio e o Dreamweaver. O Eclipse, juntamente com o PHPEclipse, JSEclipse, o EclipseTidy e o QuantumDB formaram o ambiente de desenvolvimento mais produtivo, além de inteiramente free e open-source. Com ele se desenvolve tando o HTML, o JavaScript quanto o código PHP. A característica mais marcante ao meu ver, a mais produtiva para desenvolvimento em PHP é o debugger. Sei que existe a variante pessoal, o gosto de cada um, e algumas pessoas podem preferir esta ou aquela ferramenta, dependendo de sua bagagem anterior e de sua atitude, mas para mim esta foi a melhor alternativa para trabalho com o PHP.

O PHPEclipse é formado por um conjunto de plugins para o Framework Eclipse que traz uma IDE integrada para desenvolvimento de PHP (http://www.plog4u.org/index.php/Using_PHPEclipse). Podemos utilizar o Eclipse com PHPEclipse tanto no Windows quanto no Linux, pois o mesmo é desenvolvido em Java para ambientes gráficos Desktop e roda em qualquer ambiente que suporte Java.

O Eclipse oferece um conjunto de recursos importantes para o desenvolvedor, como o CVS e Subversion, Webdav, busca texto em projetos e muitos plug-ins de terceiros, além de uma grande comunidade de usuários e desenvolvedores.

Características do PHPEclipse

• PHP, HTML, XML and CSS Syntax Highlighting

• Code completion

• Javascript Editor (via JSEclipse)

• Integrated web browser preview (somente no Windows)

• Integrated control of Apache & MySQL servers (if using XAMP distribution)

• Some PHPUnit Integration (?)

• Integrated MySQL client view using Quantum DB (PostgreSQL e outros bancos)

• HTMLTidy integration

• Debug

• Marca início e final de chaves e parênteses, colchetes e chaves

(?) Ainda não foram testados

Integrado aos plugins JSEclipse e EclipseTidy oferece ainda mais recursos.

Características da IDE Eclipse+PHPEclipse+JSEclipse+EclipseTidy:

• Colorir sintaxe.

• Auto-completar código:

• PHP – (Ctrl+BarraDeEspaço)

• HTML (digitar parte e aguardar. Exemplo: digitar "<" e aguardar ou parte e Ctrl+Espaço. Ajuda com Tabelas, forms, etc.

• JavaScript (sililar ao HTML), exemplo: digitar "<script", aguardar um pouco e teclar Ctrl+Espaço

• Tanto para HTML quanto para JavaScript ele completa com trechos de código. Se teclar <html e Ctrl+Espaço ele exibe todo o cabeçalho do HTML. Tecle "$" e aguarde, ou "$n".

Obs.: é sensível ao contexto. Só auto-completará uma função php num script php.

• Ajuda Rápida ao selecionar palavra-chave e mover mouse sobre a mesma.

• Ajuda completa ao selecionar palavra-chave e teclar Ctrl+Shift+H ou botão direito e PHP Help)

• Debugar o código logo que teclamos Enter em novo script, ou seja, a cada linha. Ao abrir script existente. Sempre que existe erro e salvamos o arquivo, o nome do arquivo e o diretório onde ele encontra-se ficam com um pequeno X vermelho acusando o erro.

• Numeração de linhas (Window – Preferences – PHPEclipse... - PHP – Show Line Numbers)

• Controle de Tabulação (Window – Preferences – PHPEclipse... - PHP – Display tab width)

• No Windows ao abrir ou ao salvar mostra o resultado na janela PHP Browser (se corretamente configurado em Window – Preferences – PHPEclipse... – Project Defaults e WebServer local).

Alterando o Help Online para Português do Brasil

(Tradução e resumo da Wikipedia sobre PHPEclipse)

• Fechar Eclipse

• Mover doc.zip de .../plugins/net.sourgeforge.phphelp... para uma pasta temporária

• Descompactar doc.zip

• Baixar o arquivo doc do idioma desejado do site oficial do php, sendo do tipo "Many HTML files"

• Descompactar no diretório doc do doc.zip, socrescrevendo todos

• Compactar o diretório doc como doc.zip (embutindo também o diretório doc). No linux:

zip -r doc.zip doc. No Windows selecionar o diretório doc e compactar com WinRar ou outro.

• Copiar doc.zip novamente para .../plugins/net.sourceforge.net.phphelp

Utilizando: Selecione uma palavra-chave em um script PHP e tecle Ctrl+Shift+H ou clique com o botão direito do mouse e PHP Help.

Instalações

Instalação do Eclipse

- Baixar a versão 3.1.1 ou superior aqui: http://www.eclipse.org/downloads/index.php preferindo um mirror brasileiro (veja abaixo na lista em South America). Requer Java (JRE ou JSDK).

- Descompacte o arquivo

Instalação do PHPEclipse

- Baixar a versão CVS 1.8 ou superior do site oficial: http://www.phpeclipse.de/

- Copie o arquivo .zip para o diretório do Eclipse. Então descompacte.

Instalação do JSEclipse

Baixe também do site:

http://www.interaktonline.com/Products/Eclipse/JSEclipse/Overview/

Instale de maneira semelhante a instalação do PHPEclipse.

Instalação do EclipseTidy

Download - http://eclipsetidy.sourceforge.net/ e instale da mesma forma dos anteriores.

Instalação do Quantum DB

Características:

• connect to databases using standard JDBC drivers

• review schemas, tables, views and sequences

• look up column, index and foreign key information

• issue ad-hoc queries or other SQL statements against the database

• manage, edit, and work with SQL files (*.sql)

• issue updates, deletes, and inserts using simple, easy-to-use wizards

Trabalha com Adabas, DB2, DB2 on AS400, HSQLDB, Informix, MySQL, Oracle, Pointbase, PostgreSQL, Sybase e vários bancos com um driver JDBC.

Download - http://sourceforge.net/projects/quantum/

Também descompacte para o mesmo diretório do Eclipse.

Para adicionar um banco, instale o JDBC correspondente a ele.

Para o PostgreSQL baixe daqui:

http://jdbc.postgresql.org/download.html (Para a versão 7.4 baixar o apropriado pacote).

Para o MySQL: http://www.aquafold.com/docs-jdbcdrivers-mysql.html

- Após descompactar abra o Eclipse e Window – Open Perpective – Other... Quantum DB e OK
- Na Perpectiva Database Bookmark (esquerda) clique com o botão direito e New Bookmark.
- Clique em Add Driver – Add External Jar... e indique o arquivo .jar que você baixou.
- Clique em Browse... e selecione o nome da classe e OK e Finish.
- Selecione o JDBC instalado e clique em Next.
- Userid – nomeuser e Password - ****** Hostname (127.0.0.1), Port (5432) Database name (banco).
- Clique em Next e digite um nome para esta conexão (sugestão: nome do banco) e clique em Next.
- Clique em Finish.
- Duplo clique no novo Bookmark que apareceu à esquerda para abrir (conectar) o banco.
- Ao selecionar e duplo clique em uma tabela, seu conteúdo é exibido abaixo e à esquerda.
- No painel de cima à esquerda podemos inserir consultas na sintaxe SQL.

Veja a tela abaixo para detalhes.

Configuração do PHPEclipse no Eclipse

Abrir o Eclipse – menu Window – Preferences

Observe que já temos JSEclipse e PHPEclipse Web Developn.

- Clique no sinal de + à esquerda de PHPEclipse

- Em Browser Configuration podemos adicionar outros navegadores ou alterar o atual. - Em PHP uma boa opção é marcar "Show line number" que exibirá os números de linha no editor. - Agora clique em PHP External Tools

Uma configuração no WindowsXP.

- Agora vejamos Project Defaults

Veja que as configurações aqui são para um servidor web rodando localmente.

- Adicionando o Help do PHP ao Eclipse (clique no sinal de + em PHP)

Indique o caminho (local) do arquivo de help do PHP no formado .chn.

A saída mais eficiente (pelos meus testes) é a de criar um novo doc.zip (como indicado).

Configurando a Atualização Automática do Eclipse

- Window – Preferences – Install/Update

Adicionar o servidor e a porta do Proxy.

Clicar em Automatic Updates e Clicar em Automacically find new updates and notify me.

Atualizando o Eclipse e seus plugins

- Help – Software Updates – Find and Install...

- Search for updates of the currently installed features

(Para atualizar o Eclipse atual e os Plugins instalados)

Selecionar um mirror e atualizar.

- Search for new features to install

(Procurar novos plugins para instalar)

Exemplo: Next – New Remote Site

Name - AnyEdit tools

URL - http://andrei.gmxhome.de/eclipse/

Criando um Novo Projeto e PHP

- File – New – Project

- PHP – Sinal de + e PHP Project e Next

- Digite um nome para o projeto e clique em Finish.

Obs.: Antes de finalizar podemos desmarcar "Use default" e salvar o projeto noutra pasta.

- Clicar sobre o nome do Projeto no painel (Navigator) da esquerda com o botão direito e New – PHP File

- Em File name informe o nome do arquivo e clique em Finish.

Recebemos então o seguinte código:

```
<?php
/*
* Created on 10/10/2005
*
* To change the template for this generated file go to
* Window - Preferences - PHPeclipse - PHP - Code Templates
*/
?>
```

Abrir uma linha em branco acima de ?> e digitar:

```
echo "Teste!";
```

Teclar Ctrl+S para salvar e observe que abaixo no painel PHP Browser já aparece o resultado:

Teste! Este recurso até o momento está disponível apenas em Windows.

Caso queira maximizar o painel PHP Browser para ver o resultado em tela cheia:

- Clique no botão maximize à direita:

Adicionando arquivos PHP ao Projeto criado

- Clique sobre o nome do projeto com o botão direito e New – PHP File.
- Em File name digite um nome para o arquivo e Finish.

Adicionando arquivos HTML e JavaScript

- Clique sobre o nome do projeto com o botão direito e New – Other – Sinal de + de PHP e HTML file.
- Em File name digite um nome para o arquivo e Finish.

Auto_completando o código

Logo abaixo da linha do echo digite "pg_" e então tecle Ctrl+BarraDeEspaços.

Veja o que aparece:

Uma relação de funções do PHP iniciadas com pg_ . A função selecionada tem uma pequena descrição aparecendo à direita.

Podemos também inserir outros arquivos no nosso projeto PHP: HTML, JS e outros formatos que também contarão com bons recursos do plugin PHPEclipse e do JSEclipse.

Debugando Scripts PHP

Procurar e corrigir erros em códigos é algo muitas vezes trabalhoso e outras vezes mais ainda. Para lidar melhor com estes problemas e resolvê-los mais rápido e eficientemente, devemos: ter um bom conhecimento dos mecanismos de correção da ferramenta em uso, da sintaxe das funções e estruturas do PHP, além de usar as melhores ferramentas de programação disponíveis: IDEs, editores, funções/classes, livros tradicionais, livros eletrônicos, o manual oficial do PHP em português do Brasil, aplicativos de exemplo e o que sobrar pesquisar no site oficial do PHP e na Internet em geral.

Algumas vezes o Interpretador acusa erro em uma linha e o erro de fato encontra-se em outra bem diferente (normalmente acima), como é o caso de strings não fechadas adequadamente e concletes, chaves ou parêntesis.

Maior atenção para os pontos:

(ponto e vírgula) ao final das instruções

aspas (") fechando com apóstrofos (') ao invés de aspas

),} ou] apenas abrindo ou fechando expressões em if com um igual (=) quando o correto são dois iguais (= =)

muita atenção com expressões lógicas

echos ou prints exibindo variáveis em pontos estratégicos. Exemplo: print "Valor $mValor
" para debugar;

Exemplo de uso do Debug do PHPEclipse

Crie um arquivo com o código abaixo:

```
<?
//include "conexao.inc.php";
//echo "Aguarde...;
/*
for ($campo=20;$campo<9020;$campo++){ $str="insert into clientes values ($campo, 'Ribamar.$campo','ribafs.$campo@gmail.com','Fortaleza.$campo', 'Ceará.$campo',$campo*2)"; pg_query($conexao,$str);
}
```

- /

```
//echo "
Concluido!";
$x=4;
if (($x ==0) && ($x>5){
echo "Furado!";
}else{
echo "Ok";
}
?>
```

Copie o arquivo para o diretório web como está. Então execute no browser.

Veja que ele retornará o seguinte erro:

Parse error: syntax error, unexpected '{' in C:\Arquivos de programas\xampp151\HTDOCS\teste.php on line 13

Não dá para identificar diretamente pela mensagem de erro, já que a { está correta.

Agora abra o Eclipse e num novo arquivo de projeto PHP cole o mesmo código acima.

Então tecle Ctrl+S para salvar. Veja que ao salvar ele grifa as duas chaves do if e adiciona dois pequenos círculos vermelhos à esquerda das linhas de número 13 e 15.

Pare o ponteiro do mouse sobre o círculo da linha 13. Receberá uma mensagem assim:

Parser error "')' expected after 'if' condition. "

Esta mensagem já nos ajuda mais que a retornada pelo browser, pois acusa de fato o problema, falta fechar o último parêntesis do if.

Configuração do PHP e Apache para melhor Debug

Isso na fase de desenvolvimento, pois na fase de produção o ideal é inibir as mensagens de erro.

Editar o arquivo php.ini e configurar:

display_errors = On

log_erros = On

error_reporting = E_ALL & ~E_NOTICE (Ativa todos, exceto os Notice)

Salvar, parar o Apache e iniciar.

Tipos de Erros e avisos do PHP

Parse error – problema com a sintaxe do script. O interpretador pára de rodar quando encontra um erro destes.

Fatal error – Um sério problema com o conteúdo do seu programa, como a chamada a uma função que ainda não foi definida. O interpretador pára ao encontrar um erro destes.

Warning – Aviso do interpretador que tem algo errado em seu programa. Exemplo: chamar função com número errado de argumentos. Mas este erro não impede de que o interpretador continue trabalhando.

Notice – Uma dica do interpretador. Exemplo: mostrar uma variável sem primeiro inicializar ela com algum valor. Não fará o interpretador parar.

Using PHPEclipse : Installation : Installing the DBG Debugger

http://www.plog4u.org/index.php/Using_PHPEclipse_:_Installation_:_Installing_the_DBG_Debugger

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/Eclipse_com_PHPEclipse%2C_Aptana_e_outros_plugins"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/PHPDesigner

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

4.2 - PHPDesigner

Site Oficial

http://www.mpsoftware.dk/downloads.php

Preferido dos visitantes do site PHP Editor Review e também deste colaborador, com versões Freeware e comercial, mas somente para Windows.

Professional is a powerful PHP Integrated Development Environment (IDE) for both beginners and professional developers that enhances the process of editing, analyzing and publishing applications and websites powered by PHP and other web languages.

Boost your productivity and simplify complex coding projects through an intelligent set of editing tools that includes full support for PHP 5.2, syntax highlighters, intelligent nested code completion, code tips that are displayed as you type, a project manager, code snippets, large built-in code-libraries, bracket/tag highlighting (matched and unmatched), and numerous tools and wizards to speed up your development.

The strength of PHP lies in its ability to embed it into, and together with, other web languages and technologies. Therefore, PHP Designer 2007 not only supports PHP, but also other web languages such as HTML, MySQL, XML, CSS, JavaScript, VBScript, Java, C#, Perl, Python and Ruby!

Support for WAMP/LAMP and AJAX developers is provided through intelligent syntax highlighting that automatically switches between syntax highlighters for PHP, HTML, CSS, and JavaScript depending on your position in the document, and dims the rest of the code!

Getting detailed information about more than 3000 native PHP functions on the fly makes the learning curve for PHP as easy and short as possible.

Jump to any declaration in both source and project files with filtering by class, interface, function, variable or constant.

Easily document your code with phpDocumentor, the standard documentation tool for the PHP language, with the built-in phpDocumentor Wizard.

The philosophy behind PHP Designer 2007 is that it be enjoyable to use for hours and hours for both beginners and professional developers through an easy-to-use, intuitive, customizable and tabbed interface. The professional edition is for any commercial, business, educational, institutional or governmental purpose!

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/Kate

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

4.3 - Kate

Editor com suporte para um grande número de linguagens programação.

Seu plugin "Auto-completar palavras" apresenta uma grande produtividade, maior até que grandes IDEs.

Site Oficial

http://www.kate-editor.org/

- Software livre, apenas para GNU/Linux (KDE)

- HighLighting da sintaxe

- Numeração de linhas

- Reconhece chaves (início e final)

- Auto-completa todas as palavras (via Plugin)

- Edição em servidores remotos via FTP e SFTP

- Class Browser (Plugin)

- Debugger Interativo (Plugin)

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/Kate"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/Dreamweaver

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

Conteúdo

- 1 Dreamweaver
- 2 Gerando Código PHP com MySQL Através do Dreamweaver
  - 2.1 1) CRIAÇÃO DO BANCO cad_clientes NO MYSQL
  - 2.2 2) DEFINIR UM SITE NO DW
  - 2.3 3) DEFINIR UM OBJETO CONEXAO
  - 2.4 4) DEFINIR O ARQUIVO DE AUTENTICAÇÃO DO USUÁRIO
  - 2.5 5) CRIAÇÃO DO MENU DE ACESSO
  - 2.6 6) CRIAÇÃO DA SEÇÃO DE CONSULTA (UM ÚNICO CLIENTE)
  - 2.7 7) CRIAÇÃO DA SEÇÃO DE CONSULTA (TODOS OS CLIENTES)
  - 2.8 8) CRIAÇÃO DA SEÇÃO DE CADASTRO
  - 2.9 9) CRIAÇÃO DA SEÇÃO DE ATUALIZAÇÃO
  - 2.10 10) CRIAÇÃO DA SEÇÃO DE EXCLUSÃO
  - 2.11 11) CRIAÇÃO DE UMA SEÇÃO MASTER/DETAILS
  - 2.12 12) ADICIONANDO O TOTAL DE REGISTROS A UMA PÁGINA
  - 2.13 13) ADICIONAR SELECT DINÂMICO A CAMPO DE FORMULÁRIO

Dreamweaver

IDE muito popular entre os designers e com suporte ao PHP, inclusive também gera código.

Site Oficial

https://store1.adobe.com/cfusion/store/index.cfm?store=OLS-US&NR=0#loc=en_us&view=ols_prod&store=OLS-US&categoryOID=1147942&distributionOID=105&NR=0

Versão comercial para Windows e MAC.

Dreamweaver 8 is the industry-leading web development tool, enabling users to efficiently design, develop and maintain standards-based websites and web applications.

Gerando Código PHP com MySQL Através do Dreamweaver

Com o Dreamweaver iremos gerar praticamente todo o código de uma aplicação, a conexão, inclusão, consultas, utentiucação de usuários, etc. Criaremos uma aplicação web para cadastro e manutenção de clientes de uma empresa. Para simplificar criaremos apenas as telas referentes aos clientes, supondo que os administradores são mantidos através de um gerenciador do banco.

1) CRIAÇÃO DO BANCO cad_clientes NO MYSQL

```
CREATE TABLE `administrador` (
  `login` varchar(8) NOT NULL,
  `senha` varchar (8) NOT NULL,
  `nome` varchar(40) NOT NULL,
  PRIMARY KEY  (`login`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE cliente (
  codigo int(11) NOT NULL auto_increment,
  nome varchar(40) NOT NULL,
  login varchar(12) NOT NULL,
  senha varchar(12) NOT NULL,
  email varchar(40) default NULL,
  login_adm varchar(8) NOT NULL,
  UNIQUE KEY login (login),
  PRIMARY KEY  (codigo),
  FOREIGN KEY (login_adm) REFERENCES administrador (login) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Observações Importantes:

- Observe que a tabela administrador tem todas as variáveis (nomes de tabela e campos) cercada por delimitadores e que estes delimitadores são sinais de crase e não apóstrofos. Veja que o uso das crases é opcional, pois a tabela clientes não os utiliza e ambas as sintaxes são aceitas.

- Para que tabelas em MySQL tenham suporte a relacionamentos, elas obrigatoriamente devem ser do tipo InnoDB (ENGINE=InnoDB ou TYPE=INNODB)

- Para dar suporte a acentuação usamos "DEFAULT CHARSET=latin1"

Comandos úteis do MySQL:

SHOW TABLES - Exibe tabelas do banco atual

SHOW DATABASES - exibe todos os bancos

DESCRIBE nometabela - mostra estrutura da tabela

2) DEFINIR UM SITE NO DW

Ao definir um site sempre prefira que as pastas locais e remotas sejam a mesma, para o caso de estar trabalhando localmente (apache e php na sua máquina). Ao final enviar para o servidor por FTP.

3) DEFINIR UM OBJETO CONEXAO

Ao definir uma conexão com o banco MySQL, teremos um objeto conexão com um nome que batizamos e este objeto estará disponível por todo o site.

- Tecle Ctrl+N e Selecione General - Dynamic page - PHP e Create

- Tecle Ctrl+S para salvar (index.php)

- Clique no painel Application - Databases (à direita).

Então clique no sinal de + (mais) e em MySQL Connection

Então preencha assim:

```
- Connection name - conexao
- MySQL server - 127.0.0.1 (caso esteja utilizando remotamente aqui entre com o
IP do servidor remoto)
- User name - root (mude se for o seu caso, deixando "" caso não tenha senha)
- Password - suasenha
- Database - cad_clientes (idealmente clique no botão select para escolher)
- Agora clique no botão Test para ver como ficou. Se tudo bem clique em OK.
- Observe que a conexão agora ganhou um ícone à direita em Databases. Se clicar
no sinal de + verá os objetos do banco.
- Para remover uma conexão basta selecionar e clicar no botão com o sinal de -
(menos)
```

Veja que no arquivo atual estando no modo Design não percebemos nada do objeto conexao criado. Mas o DW cria algumas sub-pastas na pasta do nosso site atual, contendo os códigos que irá utilizar em todo o aplicativo. Na pasta Connection está o arquivo conexao.php, que contém o seguinte código:

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
```

```
# HTTP="true"
$hostname_conexao = "127.0.0.1";
$database_conexao = "cad_clientes";
$username_conexao = "root";
$password_conexao = "phpsenha";
$conexao = mysql_pconnect($hostname_conexao, $username_conexao,
$password_conexao) or trigger_error(mysql_error(),E_USER_ERROR);
?>
```

Este código é um código simples de uma conexão em PHP com MySQL, mas quando já tiver dado os primeiros passos em PHP e também no PHP voltado ao objeto e quizer ir mais adiante estude o código do arquivo mysql.php da pasta _mmServerScripts.

4) DEFINIR O ARQUIVO DE AUTENTICAÇÃO DO USUÁRIO

Este código exige que criemos um formulário de login com os campos login e senha. Ele redireciona o usuário devidamente autenticado para um arquivo (menu.php no nosso caso) e o não autenticado para outro (index.php).

```
- Ainda com o arquivo index.php aberto e no modo DESIGN faça...
- No painel Properties abaixo clique em Centralizar e Format Heading 2
- Então digite "Autenticação do Usuário" então tecle Enter
- Crie um form contendo apenas dois campos texto login e senha e um botão
(Acessar).
- Os nomes dos campos devem ser "login" e "senha"
- O campo senha do tipo password.
- Os campos devem ter size (char width) 9 e maxlength (maschar) 8
```

- Agora usaremos o assistente do DW para gerar o código de autenticação:

- Selecione a barra de ferramentas Application (acima)

- Clique na seta ao lado do botão User authentication (último da barra)

- Selecione "Log in user"

Então preencha o assistente como abaixo:

- Validate using connection - Selecione “conexao”
- Table - administrador
- Username column - login
- Password column - senha
- If login suceeds, go to - menu.php
- If login fails, go to - index.php e marque também Go to previous URL

- Restrict access based on - Username and password. A opção abaixo é indicada para quando temos diversos administradores com diversos níveis de acesso: um acessa tudo, outro acessa somente parte, ...

Com isto também não vemos nada no modo Design, mas ao observar o modo Code vemos o código inserido pelo DW:

```
<?php require_once('Connections/conexao.php'); ?>
<?php
// *** Validate request to login to this site.
session_start();
$loginFormAction = $_SERVER['PHP_SELF'];
if (isset($accesscheck)) {
  $GLOBALS['PrevUrl'] = $accesscheck;
  session_register('PrevUrl');
}

if (isset($_POST['login'])) {
  $loginUsername=$_POST['login'];
  $password=$_POST['senha'];
  $MM_fldUserAuthorization = "";
  $MM_redirectLoginSuccess = "menu.php";
  $MM_redirectLoginFailed = "index.php";
  $MM_redirecttoReferrer = true;
  mysql_select_db($database_conexao, $conexao);

  $LoginRS__query=sprintf("SELECT login, senha FROM administrador WHERE
login='%s' AND senha='%s'",
    get_magic_quotes_gpc() ? $loginUsername : addslashes($loginUsername),
get_magic_quotes_gpc() ? $password : addslashes($password));

  $LoginRS = mysql_query($LoginRS__query, $conexao) or die(mysql_error());
  $loginFoundUser = mysql_num_rows($LoginRS);
  if ($loginFoundUser) {
     $loginStrGroup = "";

    //declare two session variables and assign them
    $GLOBALS['MM_Username'] = $loginUsername;
    $GLOBALS['MM_UserGroup'] = $loginStrGroup;

    //register the session variables
    session_register("MM_Username");
    session_register("MM_UserGroup");

    if (isset($_SESSION['PrevUrl']) && true) {
      $MM_redirectLoginSuccess = $_SESSION['PrevUrl'];
    }
    header("Location: " . $MM_redirectLoginSuccess );
  }
  else {
    header("Location: ". $MM_redirectLoginFailed );
  }
}
?>
```

Um código organizado e com muita coisa para se aprender.

5) CRIAÇÃO DO MENU DE ACESSO

Este arquivo não terá código do DW, apenas será o menu de acesso às demais seções do site e conterá um form com uma caixa de texto (login) e cinco botões como abaixo:

Consultar Um, Consultar Todos, Cadastrar, Atualizar e Excluir. Os botões darão acesso às demais seções do site. - O form terá o método POST, action ="" e cada botão terá código JS como abaixo ao final da tag <input ..> respoinsável pelo redirecionamento para a página solicitada:

```
<input name="btnConsultar1" type="submit" id="btnConsultar3" value="Consultar
Um" onClick="form1.action='consultar1.php';form1.submit()">
<input name="btnConsultar2" type="submit" id="btnConsultar2" value="Consultar
Todos" onClick="form1.action='consultart.php';form1.submit()">
<input name="btnCadastrar" type="submit" id="btnCadastrar" value="Cadastrar"
onClick="form1.action='cadastrar.php';form1.submit()">
<input name="btnAtualizar" type="submit" id="btnAtualizar" value="Atualizar"
onClick="form1.action='atualizar.php';form1.submit()">
<input name="btnExcluir" type="submit" id="btnExcluir" value="Excluir"
onClick="form1.action='excluir.php';form1.submit()">
```

TESTE DO SITE

Já podemos testar o site criado até o momento.

Abra no DW a página index.php e tecle F12 para abrir no site web.

Entre com login e senha que cadastrou no banco e clique em Acessar ou apenas tecle Enter. Veja que se realmente entrou com dados que estão no banco, tabela administrador, você será redirecionado para menu.php, caso contrário para index.php.

6) CRIAÇÃO DA SEÇÃO DE CONSULTA (UM ÚNICO CLIENTE)

- Criar um novo arquivo (consultar1.php, que servirá para consultar um único usuário).

No modo Design, centralizado digite no Format Heading2 "Consultar um Clientes" então tecle Enter.

CRIAÇÃO DE UMOBJETO RECORDSET

- À direita clique no painel Application – Bindings

- Clique no sinal de mais (+) e em Recordset (Query) para inserir um objeto Consulta:

Name: rsConsCli1

Connection: conexao (selecionar ou criar em Define)

Table: cliente (selecionar)

Columns: All

Filter: login e = Form Variable – login (Para o caso de método GET usar URL variable)

Sort: nome Ascending

Ao clicar em em Test não receberá retorno satisfatório, pois a variável vinda do form está vazia. Clique em OK para concluir.

Com isso não aconteceu nada visível no nosso arquivo mas temos agora um objeto contendo o código de uma consulta que poderá ser utilizada somente neste arquivo.

Após definir um Recordset ele fica disponível em Application – Bindings. Mesmo que não estejamos vendo nada de novo no modo Design, no modo Code podemos ver todo o seu código inserido pelo DW. Como também podemos inserir qualquer um dos campos do Recordset no nosso

documento, basta clicar em seu sinal de + à direita e ele mostra todos os seus campos. Então clique em qualquer campo e arraste para o documento. Então o DW insere o código PHP correspondente a introdução do campo já referenciado pelo Recordeset.

Veja o código do Recordset inserido no arquivo atual:

```
<?php require_once('Connections/conexao.php'); ?>
<?php
$colname_rsConCli1 = "1";
if (isset($_POST['login'])) {
  $colname_rsConCli1 = (get_magic_quotes_gpc()) ? $_POST['login'] :
addslashes($_POST['login']);
}
mysql_select_db($database_conexao, $conexao);
$query_rsConCli1 = sprintf("SELECT * FROM cliente WHERE login = '%s' ORDER BY
nome ASC", $colname_rsConCli1);
$rsConCli1 = mysql_query($query_rsConCli1, $conexao) or die(mysql_error());
$row_rsConCli1 = mysql_fetch_assoc($rsConCli1);
$totalRows_rsConCli1 = mysql_num_rows($rsConCli1);
?>
```

INSERIR UMA DYNAMIC TABLE

Agora que temos o nosso recordset vamos inserir o resultado da consulta iniciada quando o usuário clica no botão Consulta Um.

Clique na seta ao lado do botão Dynamic Date e clique em Dynamic Table:

Então preencha os dados do assistente:

Recordset – selecione o nosso REcordset criado rsConCli1

Show - 10 registros por vez

Border – 1 (para que a tabela exiba bordas)

Então teremos algo como:

7) CRIAÇÃO DA SEÇÃO DE CONSULTA (TODOS OS CLIENTES)

Vamos agora criar o arquivo consultart.php.

Vamos adicionar um Recordset com nome rsConCliT. Este Recordset receberá todos os registros da tabela sem nenhum filtro, apenas ordenados pelo nome:

Vamos também adicionar uma Dynamic Table tendo como base o rsConCliT:

INSERINDO PAGINAÇÃO DE RESULTADOS

Vamos também inserir uma paginação de resultados, pois agora moderemos ter uma grande quantidade de registros. A paginação controla a exibição dos regitros, onde selecionamos quantos registros queremos exibir de cada vez e ainda oferece links para navegar entre os registros existentes: próximos, primeiros, últimos, anteriores.

Deixe o cursor abaixo da tabela criada e clique na seta ao lado do botão Recordset Paging e clique em Recordset Navigation Bar:

Quando verá:

Apenas selecione Images e clique em OK.

Podemos deixar o resultado maia elegante centralizando a tabela criada pelo Dynamic Table: deixe o cursor do mouse abaixo da tabela até aparecer a seta e clique na tabela. Quando perceber toda a tabela estiver selecionada então aparece nas propriedades abaixo a opção Align (selecione Center e salve).

Outra sugestão é setar a propriedade Alt de todas as quatro imagens da paginação com: Primeiro, Próximo, Anterior e Último. Basta selecionar a imagem e setar abaixo na propriedade.

8) CRIAÇÃO DA SEÇÃO DE CADASTRO

Para a seção de cadastro (insert) não requer criação de Recordset nem de Formulário.

- Clique na seta ao lado do botão Insert Record

E clique em Record Insertion Form Wizard preenchendo como sugerido abaixo:

Veja que removi o campo código, que é preenchido pelo próprio MySQL.

9) CRIAÇÃO DA SEÇÃO DE ATUALIZAÇÃO

Este assistente exige que criemos apenas um recordset e depois ele criará o formulário de atualização.

- Crie um novo arquivo atualizar.php

- Em Heading 2 e centralizado digite "Atualização de Clientes" e tecle Enter

- Clique na seta ao lado do botão Update Record e clique em Record Update Form Wizard

Então aparece um diálogo exibindo as exigências do assistente, no caso ainda resta criar um Recordset:

Clique no link Recordset para criar um agora e crie de acordo com os dados abaixo:

Para testar será exigido um login de cliente cadastrado.

Clique em OK para criar e confirme OK no diálogo Setup Instructions.

Então será aberto o assistente de criação do form de atualização.

Record Update Form

Observe que removemos o campo código, já que este sempre será mantido pelo próprio MySQL (auto increment).

Outro detalhe é a página para onde será redirecionado após a atualização. Caso no site se faça muitas atualizações de cada vez o ideal seria voltar para a própria página atualizar.php.

Em casos de grande sigilo podemos alterar o tipo de exibição da senha para Password em Display as.

Altere o botão para "Atualizar Registro".

10) CRIAÇÃO DA SEÇÃO DE EXCLUSÃO

Esta também exige a criação de um Recordset e também cria automaticamente o form.

Como este assistente é somente código não requer que digitemos nenhum título. Após a exclusão ele voltará para a página menu.php.

- Clique no botão Delete Record

Então entre com as devidas informações nno assistente:

11) CRIAÇÃO DE UMA SEÇÃO MASTER/DETAILS

Este assistente cria uma página onde podemos ter uma página exibindo uma grande relação de registros e com apenas poucos campos. Um destes campos contém um link que abre uma segunda página contendo mais detalhes sobre o registro selecionado.

- Crie um arquivo novo chamado marterdet.php.

- Com o arquivo aberto crie um novo Recordset para exibir todos os campos da tabela cliente.

- Clique no botão Master Detail Page Set

Agora complete de acordo com a captura abaixo:

Observe que deixei apenas dois campos para a página master e na detail todos os campos. O DW criará automaticamente a página detail.php indicada e ainda adiciona paginação ao resultado.

Antes de testar abra a página detail.php e salve.

12) ADICIONANDO O TOTAL DE REGISTROS A UMA PÁGINA

- Após ter criado uma página que exibe diversos registros, então jogue o foco onde deseja exibir o total de registros e digite algo como "Total de Clientes Cadastrados: "

```
- Clique na seta do botão Display Records Count e clique em Total Records
```

Agora selecione o Recordset

E clique em OK

Ficando assim: Total de Clientes Cadastrados: {rsConCliT.TotalRecords}

Se verificar vai constatar que existem diversos outros recursos de geração de código do DW.

13) ADICIONAR SELECT DINÂMICO A CAMPO DE FORMULÁRIO

Podemos adicionar um select que vai a outra tabela e é preenchido com valores da mesma e armazena na tabela atual.

- Selecione o campo onde deseja adicionar o select dinâmico e remova o campo texto (login_adm) do cadastrar.php

- Em seu lugar insira um elemento do formulário List/Menu.

- Mude seu nome para login_adm.

- Crie um Recordset contendo toda a tabela administrador.

- Selecione o select e abaixo clique no botão Dynamic deixando assim:

Então clique no botão Select value equal to e selecione nome e OK:

Prontinho, temos agora um select dinâmico.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/Dreamweaver"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/Zend Studio

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

4.5 - ZendStudio

Uma das mais populares IDEs para PHP.

Comercial, com versões para Windows, Linux e Mac.

Site Oficial

http://www.zend.com/products/zend_studio

Principais Recursos

Code Completion, Code Folding, PHPDoc / PHPDocumentor Support, Code Templates, Debugger Profiler, Firewall/NAT Traversal, Proxy Settings, PHP Code Analysis, Smart Goto Source, Navigation Web Services Support (SOAP), Java, Embedded Internet Explorer, Code Snippets, SQL Support, Find in Files, Zend Platform Integration, Zend Framework Integration, Source Control, CVS Integration, Subversion Integration, FTP Advanced Integration, Help, Languages

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/Zend_Studio"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/VS.PHP

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

4.6 - VS.PHP

IDE tendo como base o Visual Studio da Microsoft, por isso VS.PHP.

Site Oficial

http://www.jcxsoftware.com/

Versão Comercial somente para Windows.

VS.Php is a Php integrated development environment based on Visual Studio. With VS.Php you can design, develop, debug and deploy Php applications within the Visual Studio IDE. VS.Php key features are around providing rich Php and Smarty editing capabilities as well as its ability to debug Php scripts locally as well as remotely.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/VS.PHP"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/Delphi for PHP

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

4.7 - Delphi for PHP

Uma das últimas IDEs a aparecer para PHP.

Versão comercial para Windows.

Site Oficial

http://www.codegear.com/products/delphi/php

The RAD visual PHP development environment

```
   * Accelerate PHP web development with Delphi's proven visual (RAD) framework
for PHP
   * Simplify PHP programming with Delphi's comprehensive PHP for VCL libraries
   * Sophisticated editing and integrated debugging speeds up the process of
tracking down bugs and errors in the code
   * Integrated development environment streamlines code management and
navigation so managing and organizing PHP code has never been easier.
```

Com a mesma facilidade do Delphi, a versão PHP permite desenvolver aplicativos para a web em ambiente 100% visual, com direito a bibliotecas de código aberto, scripts e compatibilidade com os principais banco de dados.

Essa nova plataforma de desenvolvimento promete revolucionar o desenvolvimento web, integrando o PHP ao Delphi, com um ambiente gráfico totalmente visual e ainda com um depurador, conexões inteligentes com MySQL e interbase. Ele utiliza uma biblioteca de classes do PHP5 integrada, chamada de VCL.

Os componentes VCL aparecem na barra de objetos e incluem botões, labels, check boxes, imagens, DHTML, menus, componentes de acesso da dados, objetos Flash e muito mais.

Tutorial - Uma introdução ao Delphi for PHP - http://webinsider.uol.com.br/index.php/2007/06/10/uma-introducao-ao-delphi-for-php/

Bom artigo sobre a ferramenta no iMasters - http://www.imasters.com.br/artigo/5990/php/delphi_for_php//imprimir/

Outro bom artigo - http://www.activedelphi.com.br/print.php?sid=489

Link com diversos vídeos - http://www.devmedia.com.br/articles/listcomp.asp?keyword=dforphp

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/Delphi_for_PHP"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/KDevelop

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

4.8 - KDevelop

O KDevelop é uma IDE for Linux (KDE) com suporte a diversas linguagens de programação, inclusive ao PHP.

Vejamos algumas dicas de como utilizá-lo com PHP.

Passos para a instalação no Linux Ubuntu (versão 6.0.6).

Requisitos

```
- Vamos adicionar todos os repositórios:

- Abrir o Synaptic - Configurações - Repositórios

- Em Mídia de instalação marque todos

- Clique em Adicionar, marque os 4 componentes e novamente Adicionar

- Clique em Adicionar novamente e agora selecione o segundo Canal e os 4
componentes e Adicionar. Repita para os 4 canais então clique em Fechar.

- Clique no botão Recarregar para atualizar os repositórios e feche o Synaptic.
```

Instalar o Kdevelop para PHP

```
- Clique no botão Aplicações - Adicionar/Remover

- Selecione o grupo Programação à esquerda

- Em aplicações acima selecione KDevelop: 'Scripiting'

- Clique em Aplicar e novamente Aplicar e então em OK
```

Configurações

Configurando para nossa acentuação:

- Tools - Encoding - Western European - ISO-8859-1

Configurando para PHP

- Tools - Highlighting - Scripting - PHP (HTML)

Configurações Diversas no Editor

- Setting - Configure Editor - Editing - Tab width - 4
- Open/Save - Encoding - Western European (ISO-8859-1)
- Highlighting - Highlight - Scripts/PHP (HTML)

Usando

- Abra o KDevelop Scripting que agora encontra-se no grupo Programação

Preparar o KDevelop para trabalhar com PHP

- Clique em Project - Project Options
- À esquerda clique em PHP Specific

Em General - "Parser PHP &Include Path" indique "/opt/lampp/bin/php-5.1.4" em sendo seu PHP instalado via Xampp, caso contrário veja em seu php.ini qual o include_path.

- Na aba Invocation, marque "Using a existing webserver".
- Na aba Webserver indique http://localhost/cadclientes/ ou outro, se for seu caso.
- Na aba Shell, em PHP Executable indique "/opt/lampp/bin/php-5.1.4", ou outro, se for seu caso.

```
Em PHP ini File indique "/opt/lampp/etc/php.ini".

Caso utilize indique "Load Zend Extension".
```

À esquerda clique em Doxygen. À direita em Output language selecione Portuguese.

Em Input indique o diretório onde irá armazenar os arquivos.

Criando um Projeto

- Clique em Project - New Project
- Expanda a pasta PHP e selecione Simple PHP Script
- Em Application name digite um nome para a aplicação (Exemplo: cadclientes)

- Clique em Next
- Em Project Options clique em Next
- Em version control system clique em Next
- Em template clique em Finish
- Vamos testar e clique no botão Run (aquele azul com uma roldana) ou tecle F9.
- Será exibida a mensagem Hello World!, como resultado do script gerado pelo KDevelop.

Adicionar Novo arquivo ao Projeto

- Clique em File - New ou Tecle Ctrl+N para abrir o diálogo "New File".
- Clique em Blank File e selecione PHP.
- Digite o nome do arquivo (Ex.: index.php)
- Deixe marcada a checkbox "Add to project".
- Abra o arquivo criado com File - Open ...

Importando Projetos PHP

- Project
- Import Existing Project - Project Type - PHP Application

Criação de Snippets

Muito útil para armazenar dicas, funções e classes.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/KDevelop"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/PSPAD

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

Site oficial - http://www.pspad.com/en/

Editor para vários tipos de linguagens: HTML, PHP, Pascal, JScript, VBScript, MySQL, MS-Dos, Perl,...

Com várias boas características:

```
   * do you work with various programming environments ?
   * do you like highlighted syntax in your source code ?
   * do you need a small tool with simple controls and the capabilities of a
mighty code editor ?
   * are you looking for a tool that handles plain text ?
   * do you want to save money and still have the functionality of professional
products ?
   * then PSPad is for you.
```

```
   * work with projects
   * work with several documents at the same time (MDI)
   * save desktop sessions to later reopen all session files
   * FTP client - edit files directly from the web
   * macro recorder to record, save and load macros
   * search and replace in files
   * text difference with color-coded differences highlighted
   * templates (HTML tags, scripts, code templates...)
   * installation contains templates for HTML, PHP, Pascal, JScript, VBScript,
MySQL, MS-Dos, Perl,...
   * syntax highlighting according to file type
   * user-defined highlighters for exotic environments
   * auto correction
   * intelligent internal HTML preview using IE and Mozilla
   * full HEX editor
   * call different external programs for different environments
   * external compiler with output catcher, log window and log parser for an
"IDE" effect in every environment
   * color syntax highlight printing and print preview
   * integrated TiDy library for formatting and checking HTML code, conversion
to CSS, XML, XHTML
   * integrated free version of the top CSS editor TopStyle Lite
   * export with highlight to RTF, HTML, TeX format to file or clipboard
   * column block select, bookmarks, line numbers, ...
```

```
   * reformat and compress HTML code, tag character case change
   * line sorting with ability to sort on defined columns and drop duplicates
   * ASCII chart with HTML entities
   * Code explorer for Pascal, C/C++, INI, HTML, XML, PHP and more in
development
   * spell checker
   * internal web browser with APACHE support
   * matching bracket highlighting
   * ...(many more features, too numerous to list)
```

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/PSPAD"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/Quanta

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

O Quanta+ é uma IDE para desenvolvimento de projetos de Sistemas de Aplicação que, através de sua configuração, permite seu ajuste a muitos modos de interação com várias linguagens de programação.

Seu universo de execução é o sistema operacional Linux, e está disponível para muitas distribuições. É comum encontrá-lo como pacote padrão de Fedora, Mandriva, RedHat, Ubuntu, Kurumin e muitas outras.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/Quanta"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/BlueFish

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

Editor BlueFish for Linux

Um editor web for Linux com support a PHP e a diversas outras linguagens de programação.

Site oficial - http://bluefish.openoffice.nl/index.html

Disponível como binário para a maioria das distribuições.

Suas características - http://bluefish.openoffice.nl/features.html

Bluefish - um poderoso editor para web designers

Bom tutorial em português sobre o BlueFish - http://www.vivaolinux.com.br/artigos/impressora.php?codigo=1079

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/BlueFish"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/PHPEdit

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

IDE com capacidade de debug remoto.

Possui sintaxe para JavaScript, PHP, HTML. Ha uma caracteristica interessante: Ele faz o highlight apenas da linguagem sendo selecionada deixando as demais em cinza claro de forma a destacar aquilo que se está editando como foco.

Ferramenta leve que é uma alternativa mais barata que o pacote da Zend.

http://www.waterproof.fr/products/PHPEdit

Esta página é somente um esboço. Expandindo-a *você ajudará a melhorar o Wikilivros.*

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/PHPEdit"
Page categories: Esboço

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/Geany

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

Geany

IDE leve em GTK2 com suporte a diversas linguagens de programação, inclusive o PHP, com versões para Linux, Windows e FreeBSD.

Site oficial - http://geany.uvena.de/

Geany is a small and lightweight integrated development environment. Geany was developed to provide a small and fast IDE, which has only a few dependencies from other packages.

Another goal was to be as independent as possible from a special Desktop Environment like KDE or GNOME. So it is using only the GTK2 toolkit and therefore you need only the GTK2 runtime libraries to run Geany.

Here are some key features of "Geany":

```
· syntax highlighting
· code completion
· auto completion of often used constructs like if, for and while
· auto completion of XML and HTML tags
· call tips
· many supported filetypes like C, Java, PHP, HTML, Python, Perl, Pascal
· symbol lists
```

What's New in This Release:

Notes for existing users: · Tab is now used for construct completion (for, if, etc.), but it is configurable with the new 'Complete construct' keybinding.

· Template files are now stored in ~/.geany/templates/ and the 'template.' filename prefix is no longer used. You will need to move any custom template files you have.

· Inserting a file header is now optional for filetype templates. Use the string '{fileheader}' to mark where the file header should be placed.

· Drag'n'Drop of text inside the editor widget will now move the text instead of copying it.

Bugs fixed:

```
· Fix segfault when pressing Ctrl-Enter when there are no workspace tags.
· Remove error indicators in all documents when linking (#1705374).
· Sort symbol list tags also by line number (#1703575).
· Fix #1717418, Hang on SQL file load.
· Fix #1718532 - Crash when opening a special HTML file.
· Add workaround for PHP closing brace de-indenting.
```

· Fix reloading of read-only documents.

Project Management:

· Add keybinding to show project properties dialog.
· Add project Run command support.
· Run Make All and Make Custom from the project base directory.

Custom Global Tags:

· Update C global tags for GTK+ 2.10 and it's dependencies.
· Add option --generate-tags (-g) to generate a global tags file from a list of source files (see docs).
· Load global tag files stored in ~/.geany/tags at startup #.
· Add Load Tags command in the File menu #.
· # This is not supported for Pascal, PHP or LaTeX files yet.

Calltips (for C-like files):

· Show up and down arrows when there are multiple calltip matches.
· Show classname in calltips.
· Parse pointers in function return type.
· Add calltip support for D constructors.

Other changes:

· Parse 'Entering directory' Make messages so opening files from error messages works for subdirectories (thanks to Josef Whiter).
· Make Go to Tag Definition/Declaration work for all tags.
· Support filetype templates for all filetypes (see docs).
· Make file header optional for filetype templates.
· Add 'Find Selected' and 'Find Prev Selected' search commands and keybindings (thanks to Jeff Pohlmeyer).
· Add Mark button to the Find dialog, and a Remove Markers item to the Document menu.
· Add 'Recurse in subfolders' and 'Extra options' checkboxes to the Find in Files dialog.
· Add 'Switch to last used document' keybinding (Ctrl-Tab).
· Add Goto Previous/Next Marker keybindings (Ctrl-, and Ctrl-.).
· Add Toggle Marker keybinding (Ctrl-M).
· Add keybinding for construct completion, and set the default to Tab.
· Add MimeType associatiations for: C++ header, Pascal, Perl, Python, httpd-PHP and XML files (thanks to Iñaki Rodriguez).
· Add brace indenting support for Perl and TCL.
· Make backspace unindent when using spaces for indentation.
· Wrap notebook pages when switching tabs.
· Speed up loading multiple C-like files slightly.
· New filetypes: JavaScript, Lua and Haskell.
· Set several widget names to allow users to define custom styles in .gtkrc-2.0.
· Add context actions to run custom commands on current selection or the current word below cursor.
· Add different auto indention modes.
· Improve replacing in rectangle selections.
· Add custom commands to send selected text through some definable commands and replace the selection with the output.
· Add command line option --column to allow setting the initial column for the first opened file on command line.
· Improve the auto scrolling of documents.
· Improve loading of the VTE library.

```
· Add an option for using spaces or tabulators when inserting some whitespace.
· Add an option to disable Drag'n'Drop in the editor widget.
```

Documentation:

· Add Project Management, Global Tags, Construct Completion sections.

· Add Bookmarks section (thanks to John Gabriele).

· Update Filetype Templates, Search sections.

Internationalisation:

· New translations: bg.

· Updated translations: ca, cs, de, es, fr, zh_CN.

Opinião em Blog sobre o Geany

http://blog.ogmaciel.com/?p=314

Geany e DevCPP, IDE's leves e gratuitas

http://claudiosvirgens.wordpress.com/2006/06/08/geany-e-devcpp-ides-leves-e-gratuitas/

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/Geany"
Page categories: PHP

Aplicativos em PHP/Editores e IDEs para PHP, CSS, JavaScript e HTML/NotePad++

De Wikibooks

< Aplicativos em PHP | Editores e IDEs para PHP, CSS, JavaScript e HTML

Notepad++

Site Oficial

Notepad++ é um editor livre do código de fonte (e recolocação do Notepad), que suporte diversas línguas de programação, funcionando sob o ambiente de MS Windows.

Este projeto, baseado no Scintilla edita o componente (um componente muito poderoso do editor), escrito em C++ com win32 puro api e STL (que assegura a velocidade mais elevada da execução e o tamanho menor do programa), está sob a licença do GPL.

Este projeto é maduro. Entretanto, mas há ainda alguns erros e características à desejar.

Estão aqui as características de Notepad++:

- Destaque da sintaxe
- Línguas suportadas(principais):

C C++ Java C# XML HTML PHP CSS Javascript ASP VB/VBS SQL Pascal Perl Python Lua TeX Ruby Lisp

- WYSIWYG:

Se você tiver uma impressora colorida, seu código fonte (ou o que quer que você quer) tera a cor apresentada no editor.

- Usuário - destaque definido da sintaxe

Permite que o usuário defina sua própria língua: não somente a sintaxe que destacam as palavras-chave, mas também os comentários e os operadores.

- Auto-completar
- Para a maioria de línguas suportadas, o usuário pode fazer sua própria lista do API.
- Multi-Linguagem
- Multi-Exibição
- Gravação e playback de Macro

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Editores_e_IDEs_para_PHP%2C_CSS%2C_JavaScript_e_HTML/NotePad%2B%2B"

Aplicativos em PHP/Referências/Windows

De Wikibooks

< Aplicativos em PHP | Referências

Conteúdo

- 1 Windows na Wikipedia
- 2 Instalação do Windows XP Professional SP2 em Português do Brasil
  - 2.1 Criar antes uma partição para backup
  - 2.2 Particionamento
  - 2.3 Baixar os seguintes softwares antes de instalar o windows:
  - 2.4 Recomendação
  - 2.5 Instalação
  - 2.6 Concluir
  - 2.7 Configurações
  - 2.8 Otimizações de Desempenho e Segurança
  - 2.9 Instalando Alguns Dispositivos não instalados
  - 2.10 Algumas teclas de Atalho úteis
  - 2.11 Utilitários Importantes

Windows na Wikipedia

http://pt.wikipedia.org/wiki/Windows

Microsoft Windows é uma popular família de sistemas operacionais criados pela Microsoft, empresa fundada por Bill Gates e Paul Allen.

O Windows é um produto comercial, com preços diferenciados para cada uma de suas versões, embora haja uma enorme quantidade de cópias ilegais instaladas, ele é o sistema operacional mais usado do mundo.

Apesar do sistema ser conhecido pelas suas falhas críticas na segurança e como plataforma de vírus de computador e programas-espiões (spywares), o impacto deste sistema no mundo atual é simplesmente incalculável devido ao enorme número de cópias instaladas.

Instalação do Windows XP Professional SP2 em Português do Brasil

Criar antes uma partição para backup

Uma boa recomendação é para se criar pelo menos duas partições, uma para o Windows e outra para backup.

Caso se tenha um HD de 80 GB, sugiro 25 ou 30 GB para a partição do Windows e o restante para Backup.

Caso queira instalar também o Linux veja a sugestão:

swap - 512MB ou 1GB / - 20GB Windows - 20 Backup - Restante

Obs.: Sempre instale primeiro o windows e depois o Linux, pois ao ser instalado o Windows sobrescreve a área de boot (MBR) o que nos deixa sem acesso direto ao Linux.

Particionamento

Uma boa opção gratuita de programa que redimensiona partições é o GParted (Gnome Partition Editor), que oferece inclusive um live CD, que vem com o Linux Slackware e o GParted.

http://gparted.sourceforge.net/

Live CD - http://gparted.sourceforge.net/livecd.php

Baixar os seguintes softwares antes de instalar o windows:

- Firewall Zone Alarme - http://www.zonelabs.com/

http://www.zonelabs.com/store/content/company/products/znalm/freeDownload.jsp

- Clamav for Windows - http://www.clamwin.com/
- Spybot - http://www.safer-networking.org/en/download/index.html
- Editor, corretor do Registro do Windows - sugestão MV RegClean -

http://www.velasco.com.br/ ou

http://superdownloads.uol.com.br/download/i17746.html

- Atualmente o Google está oferecendo diversos utilitários muito bons, como o Norton Security Scan (Detecta e elimina vírus e worms da Internet e Atualizações gratuitas da detecção e verificação programada), como também o Spyware Doctor Starter Edition (Utilitário Anti-spyware Detecta/rem. spyware, adware, cavalos de Tróia e keyloggers, Inclui Smart Updates e programador para proteger o seu PC) além de outros bons utilitários. Caso seu micro tenha 512 MB de RAM e Internet banda larga prefira usar o Norton do Google e também o Spyware Doctor. Endereço:

http://pack.google.com/intl/pt-br/pack_installer_required.html?hl=pt-br&gl=br

- Br Office (Office grátis) - www.openoffice.org.br

- FireFox (navegador grátis) - www.mozilla.org

- ThunderBird (cliente de e-mail grátis) - www.mozilla.org

- InkSkape - Editor gráfico profissional gratuito - http://www.inkscape.org/

Recomendação

: Caso seja a primeira vez que realiza uma instalação do Windows, leia com atenção as recomendações do instalador.

Instalação

1 - Dar boot com CD do windows no drive e com boot pelo CD habilitado

2 - Fique atento e quando solicitado pressione qualquer tecla para iniciar pelo CD e aguarde... Observe as mensagens para instalação de outros dispositivos (F6 para instalar SCSI por exemplo)

3 - Bem vindo

```
Tecle Enter
```

4 - Licença Tecle Enter

5 - Selecionar partição NTFS ou FAT32 Caso já tenha uma partição com windows instalado, pressione Esc para instalar novamente sobre a partição existente.

6 - Selecione a partição onde deseja instalar e tecle D para deletar

7 - Tecle Enter para continuar a exclusão

8 - Tecle L para realmente excluir

9 - Selecione o espaço escluído e tecle C para criar uma nova partição

10 - Tecle enter para criar com todo o espaço ou entre com o tamanho e Enter

11 - Selecione a partição criada e tecle Enter

12 - Formatar com NTFS (recomendada), Enter e aguarde...

13 - Aguarde a cópia dos arquivos para a instalação...

14 - Deixe reiniciar agora pelo HD e aguarde a instalação ...

15 - Opções regionais e idioma

Clique em Avançar e opós a instalação faça os ajustes necessários

16 - Digite nome, organização e Avançar

17 - Entre com a chave e Avançar

18 - Entre com:

- Nome do computador (importante se em rede)

- Senha do administrador

- Confirme e avançar

19 - Ajuste fuso horário, data e hora e Avançar e aguarde...

20 - Configurações de rede

- Caso esteja em rede e seja necessário, clique em Configurações personalizadas e entre com as informações

- Caso contrário clique em Avançar

21 - Domínio em grupo

- Entre com dominio se em rede

- Caso contrário apenas clique em Avançar e aguarde... (o micro irá reiniciar pela primeira vez)

22 - Bem-vindo

Clique em Avançar

23 - Caso tenha pouca memória e/ou prefira um melhor desempenho clique em Não agora

Caso contrário clique na primeira opção e Avançar

24 - Como este computador se conecta à Internet?

Marcar ou ignorar

25 - Registro

Selecionar e Avançar

26 - Usuário(s)

Digite pelo menos um e Avançar

27 - Obrigado

Concluir

Barreiras de Proteção para Suprir as Deficiências de Segurança do Windows

Antes de sua primeira conexão à Internet, instale os programas sugeridos no início.

Após a instalação, reinicie o micro e então os atualize em sua primeira conexão.

Configurações

Mudar Resolução para 1024 X 768

- Clique com o botão direito numa área livre da tela - Propriedades - Configurações

- Em Resolução da tela arraste para 1024 X 768 ou outra desejada

- Clique em OK e confirme

Alterar menu iniciar para o tipo clássico (aquele usado pelo Windows 98)

- Clicar com botão direito sobre botão Iniciar - Propriedades

- Clicar em "menu iniciar clássico"

Exibir Internet Explorer e outros na Barra de Tarefas

- Clicar com botão direito sobre botão Iniciar - Propriedades - Aba Barra de Tarefas

- Mostrar barra "Inicialização Rápida"

Otimizações de Desempenho e Segurança

- Veja o tutorial do Carlos Morimoto em seu site, seção Tutoriais:

http://www.guiadohardware.net/tutoriais/049/print.php

Obs.: Tome cuidado para não desabilitar serviços importantes.

Instalando Alguns Dispositivos não instalados

- Vá ao painel de controle - Sistema - Hardware - Gerenciador de Dispositivos

- Remova todos os itens com uma interrogação amarela e tenha o CD com o respectivo driver em mãos

- Insira o CD com o driver do dispositivo e o instale

Algumas teclas de Atalho úteis

B = Bandeirinha do windows

B + E - Abrir o windows explorer

B + M - Minimiza todas as janelas abertas

B + U - Abre o Gerenciador de utilitários para portadores de deficiência

Ctrl+Esc - Abre o menu iniciar

Alt+Tab - Alterna entre as janelas abertas

Utilitários Importantes

- Desfragmentação

- Recuperação do sistema (Acessórios - Ferramentas de Sistema)

Em caso de algum problema de difícil solução podemos com esta ferramenta "voltar o tempo" para antes da ocorrência do problema e assim resolver de forma mais cômoda o problema.

Apenas o hardware e software instalados serão removidos, os arquivos criados ou copiados não serão afetados e a operação é reversível, como também podermos trocar para outro ponto de recuperação.

Em caso de problema pode ser útil, mas o ideak é sempre realizar backup dos seus dados importantes.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/Windows"
Page categories: PHP

Aplicativos em PHP/Referências/Linux

De Wikibooks

< Aplicativos em PHP | Referências

Conteúdo

- 1 Linux na Wikipedia
- 2 Instalação do Ubuntu 6.06
  - 2.1 Instalando o Ubuntu no HD com Windows
- 3 Instalação do GParted (particionador)
- 4 Dicas sobre o Linux Ubuntu
  - 4.1 Adicionar diretório ao PATH
  - 4.2 Instalando Driver ATI e Nvidia no Ubuntu com o Envy
  - 4.3 Tecla Windows no gnome e Outras teclas de atalho
- 5 Guia do ubuntu Linux
- 6 Configurando um servidor de email com Postfix e outros
- 7 Dicas Sobre Pendrives, MP3, Câmeras e cia
  - 7.1 Slax no Pendrive
  - 7.2 Formatando e Consertando o Pendrive no Linux
  - 7.3 Recuperando dados e fotos de câmeras digitais e pen drives
  - 7.4 Formatar Pendrive
  - 7.5 Instalando o Slax no pendrive. Fácil, funcional e corrigido
- 8 Permissões - referência rápida
- 9 Sistema de Vigilância Digital com ZoneMinder
- 10 Gravar CD de música pelo K3B
- 11 Evitando gargalos de hardware
- 12 VirtualBox Opção Livre para Virtualização
  - 12.1 Tutorial de uso
  - 12.2 Virtualizando com VirtualBox
- 13 Rodando o Linux dentro do Windows "nativamente"
  - 13.1 Fazendo uma conexão bridge no VirtualBox
- 14 Personalizando a Tela de Login do Ubuntu 7.04
- 15 Instalando sem complicações o Java no Ubuntu 6.06, 6.10 e 7.04
- 16 Guia de Inicialização de serviços no Linux - parte I
- 17 Multimídia
  - 17.1 Assista vídeos com suas suas músicas preferidas
- 18 Dicas sobre Desempenho
  - 18.1 P Q P, quem puxou o freio de mão?
- 19 Dicas Diversas
  - 19.1 Listando diretórios e arquivos
- 20 Shell Scripts
  - 20.1 Programando em shell-script (por Hugo Cisneiros)

Linux na Wikipedia

http://pt.wikipedia.org/wiki/Linux

Linux refere-se a qualquer sistema operacional do tipo Unix que utiliza o núcleo Linux. É um dos mais proeminentes exemplos de desenvolvimento com código aberto e de software livre. O seu código fonte está disponível sob GPL para qualquer pessoa utilizar, estudar, modificar e distribuir livremente.

GNU/Linux refere-se a qualquer sistema operacional do tipo Unix que utiliza o núcleo Linux e também os programas de sistema GNU. Como os casos de sistemas de núcleo Linux sem os programas de sistema GNU são raros, freqüentemente GNU/Linux e Linux são sinônimos.

Inicialmente desenvolvido e utilizado por nichos de entusiastas em computadores pessoais, o sistema Linux passou a ter a colaboração de grandes empresas, como a IBM, a Sun Microsystems, a Hewlett-Packard, e a Novell, ascendendo como principal sistema operacional para servidores -- oito dos dez serviços de hospedagem mais confiáveis da Internet utilizam o sistema Linux em seus servidores web.

Linux tornou-se o sistema capaz de funcionar no maior número de arquiteturas computacionais possíveis. É utilizado em aparelhos variando desde supercomputadores, até celulares, e vem ganhando popularidade no mercado de computadores pessoais.

Instalação do Ubuntu 6.06

Esta instalação refere-se à versão 6.06 do Ubuntu mas é semelhante às versões posteriores, pelo menos até a versão 7.04:

Nesta instalação considera-se que já temos o windows instalado na máquina, o que representa alguma diferença.

Alerta: no caso de se ter dados importantes no HD, quando se deve fazer backup em CD ou de outro tipo de mídia, pois no momento do particionamento sempre se corre o risco de excluir acidentalmente e indevidamente alguma ou todas as partições.

Caso não existisse o Windows, uma sugestão para um HD de 80 GB seria:

1 - swap (512 MB)

2 - / (raiz, 20 GB)

3 - /home (restante)

A partição home funciona como backup e sempre que se precisar formatar a raiz, preservar a home.

Instalação

1 - Dar boot com CD no drive (com boot pelo CD habilitado)

2 - No primeiro menu texto que aparece apensa tecle Enter e aguarde

3 - Quando entrar no ambiente gráfico e estiver pronto, clique no botão Install

4 - Idioma - Selecionar e avançar

5 - Fuso Horário - Selecione com o mouse e avance

6 - Teclado - Selecione e avance

7 - Usuário - Entre com nome, login, senha e nome do computador (importante somente se numa rede) e avançar

8 - Tipo de Instalação

- Editar manualmente a tabela de partições

Observação - Para HD virgem escolha a primeira opção

9 - Preparar Partição

Caso queira rdimensionar alguma partição, selecione-a e com o botão direito escolha redimensionar

Avançar

10 - Preparar Pontos de Montagem

Caso seu HD tenha partição que deseja preservar fique muito atento nesta etapa, especialmente deverá identificar as partições que não deseja reformatar.

Caso esteja em dúvida:

- Acesse o segundo terminal com "Ctrl+Alt+F2"

- Digite: "sudo fdisk -l"

- Anote o número das partições

- Volte para o terminal gráfico com "Ctrl+Alt+F7"

Somente após escolher com segurança os pontos de montagem marque os checkboxs Reformatar.

Caso seu HD seja virgem ou não tenha nada para preservar e não saiba o que fazer, uma sugestão é:

1 - swap (512 MB)

2 - / (raiz, 20 GB)

3 - /home (restante)

11 - Pronto para Instalar

Mais uma oportunidade para checar as partições que serão reformatadas.

Verifique e em caso de engano volte e corrija.

Clique em Install e aguarde...

12 - Instalação Finalizada

Clique em Reiniciar agora

Quando o CD for ejetado, remova- e tecle Enter

13 - Logo no primeiro login (se conectado à Internet) poderá realizar as atualizações clicando no pequeno botão laranja.

14 - Realização de algumas configurações restantes:

- Sistema - Administração - Rede

15 - Instalar alguns programas extras:

Adicionar/Remover (lembre de marcar os dois checkbox: Instalar programas não suportados e Instalar programas ...)

K3B - Gravação de CD/DVD

KolourPaint

Instalando o Ubuntu no HD com Windows

http://andregondim.eti.br/?p=21

Instalação do GParted (particionador)

Tutorial Passo-a-passo de Uso do Gparted

O Gparted é um particionador free que oferece um liveCD para esta função.

Suporta diversos tipos de sistemas de arquivos: ext2, ext3, raiserfs, ntfs, fat32, etc.

A distribuição do liveCD é o Slackware e o iso tem apenas 27,9MB.

Site oficial - http://gparted.sourceforge.net/

Live CD - http://gparted.sourceforge.net/livecd.php

Aqui vou considerar que tenho dois HDs e pretendo excluir todas as partições de um e criar uma única.

O primeiro HD está como master na porta primária, portanto o hda no Linux.

O segundo HD está como master na secundária, portanto hdc. Muito cuidado no momento de excluir as partições, para selecionar o HD correto.

1 - Dar boot com o liveCD

2 - Ao surgir o prompt apenas tecle Enter

3 - Language

Selecione portuguese e Enter

4 - Keymap

Selecione querty/br-abnt2.map para tipo ABNT2

5 - Display depth

Apenas Enter

6 - XRES

Enter e aguarde...

7 - Acima e à direita existe uma combo contendo os HDs existentes. Por default aparece o hda. No meu caso vou selecionar o hdc.

- Esta é a etapa em que também podemos redimensionar alguma partição. Para isso apenas selecionamos a partição e clicamos sobre a mesma com o botão direito e Resize/Move.

Então clicamos na seta para a direita e arrastamos até o tamanho desejado. E clicamos em Aplicar.

- Muita atenção nesta etapa para realmente selecionar o HD correto.

- Selecionar cada uma das partições a Apagar e clicar em Apagar.

- Caso tenha partições extendidas, antes deve apagar todas as suas partições lógicas e só então excluir a extendida.

- Após excluir todas clicar em Aplicar, confirmar e Fechar.

- Então selecionar o espaço "unallocated" e clicar em Novo.

- Podemos informar o tamanho da nova partição ou aceitar o total

- Podemos também selecionar o tipo de sistema de arquivos

- Finalmente clicar em Aplicar, confirmar e Fechar

8 - Clicar no menu GParted e Sair

9 - Clicar no botão vermelho do canto direito

10 - Selecionar a opção Eject e Reboot e OK

11 - Caso não ocorra a reinicialização pressione o botão de reset.

Pronto, seu HD está particionado.

Dicas sobre o Linux Ubuntu

Adicionar diretório ao PATH

Encontrada numa lista sobre Ubuntu.

Sobre a variável PATH, se ela não está em /etc/profile ou está em /etc/bash.bashrc ou em /etc/environment. Considerando que você use bash como shell. Se você quiser mudar só para um usuário veja os arquivos ~/.bash_profile e ~/.bashrc.

Abraço, Gustavo Noronha Silva <kov em debian.org>

Instalando Driver ATI e Nvidia no Ubuntu com o Envy

http://andregondim.eti.br/?p=138

Estas dicas também se aplicam à quase todas as distribuições baseadas na distribuição Debian.
Site oficial do Ubuntu - http://www.ubuntulinux.org

Formatar Pendrive
sudo umount /dev/sda1
sudo mkdosfs -F 32 -n Tiago /dev/sda1 (Tiago é o nome do volume)
Ou podemos usar o fdisk para remover e criar novamente a partição

Usando apt - Gerencia pacotes dos repositórios (instala, remove, etc)

Instalar pacote, inclusive dependências. Podemos usar * no nome:
sudo apt-get install nomepacote

Atualiza lista de repositórios
sudo apt-get update

Atualiza todos os pacotes da distribuição
sudo apt-get upgrade

Atualiza todos os pacotes da distribuição para uma nova versão
sudo apt-get dist-upgrade

Desinstalar pacote
sudo apt-get remove nomepacote

Remover pacote e seus scripts de configuração
apt-get --purge remove nomepacote

Instalar xorg e dependências
sudo apt-get build-dep xserver-xorg

Instalando dependências
sudo apt-get install -y nomedependeicia1 nomedependeicia2

Corrigir Falhas nos Arquivos Instalados

sudo apt-get install -f

Procurar ocorrências de nome nos repositórios

sudo apt-cache search nome

Exibir informações sobre o pacote

sudo apt-cache show package

Exibir informações sobre as dependências do pacote

sudo apt-cache depends package

Instalar GCC e cia

sudo apt-get install build-essential

sudo apt-get install g++-3.4 export CC=/usr/bin/gcc-3.4 sudo apt-get install linux-headers-$(uname -r)

Suporte a Java no Mozilla

sudo apt-get install j2re1.4-mozilla-plugin

Suporte a Flash no Firefox

sudo apt-get install swf-player

Instalar fontes do Java

sudo apt-get source java-package

Remover configurações e remover o pacote

sudo apt-get --purge remove nomepacote

Atualizar Firefox no Ubuntu para pt-br

sudo apt-get install mozilla-firefox-locale-pt-br

Remover arquivos não mais necessários

sudo apt-get autoremove

Usando dpkg

Instala pacotes localmente. Antes devemos ter feito download dos pacotes .deb

dpkg -i nomepacote.deb // Instalar pacote

dpkg -r nomepacote.deb // Desinstalar pacote

dpkg-deb --contents nomepacote.deb // Exibir o conteúdo de um pacote

dpkg-deb -x nomepacote.deb // Extrai o conteúdo de um pacote

dpkg-reconfigure xserver-xorg //RECONFIGURAR PLACA DE VÍDEO (ambiente gráfico)

dpkg --purge nomepacote // Remove inclusive scripts de configuração

dpkg --search nomearquivo // Procurar em que pacote está um arquivo

URL sobre dpkg - http://pt.wikipedia.org/wiki/Dpkg

Repositório de todos os pacotes do Ubuntu, com pacotes .deb (muito bom):

http://packages.ubuntulinux.org/

- Digite o nome do pacote em Keywork
- Clique em Search
- Clique no link mais adequado à sua busca
- Role a tela e clique na sua arquitetura. Algumas vezes All.
- Clique agora num dos espelhos de pacotes para fazer o download.

Criação de Disquete de Boot

fdformat /dev/fd0

ou

mformat a:

dd if=bootfloppy-usb.img of=/dev/fd0 bs=8192

dd if=boot.img of=/dev/fd0 bs=1024 conv=sync ; sync

dd if=bootfloppy-usb.img of=/dev/fd0 bs=36b

Criando o disco de boot automaticamente

grub-floppy /dev/fd0

Manualmente

cd /lib/grub/i386-pc

dd if=stage1 of=/dev/fd0 bs=512 count=1

dd if=stage2 of=/dev/fd0 bs=512 seek=1

Converter utf-8 para iso8859-1

No linux

iconv -f utf-8 -t iso8859-1 arquivo

Reinstalando o GRUB

grub-install /dev/hda

Identificando Distribuição

dmesg | head -1

sudo cat /proc/version

cat /etc/issue

Dicas do Fábio B. Pinto para as ótimas Dicas-L

Multimídia

Conversor de Vários Formatos de Vídeo/Som

sudo apt-get install soundkonverter

Converter Wav para MP3

sudo apt-get install audacity

For windows free - http://www.vuplayer.com e http://www.mediamonkey.com/download.htm

HOW-TO rodar Adobe Photoshop CS2 on Ubuntu

http://blog.publicidadpixelada.com/2006/10/10/how-to-adobe-photoshop-cs2-on-ubuntu-10-steps/

VLC - Player com diversos codecs

apt-get install vlc

apt-get install mozilla-plugin-vlc

K3B com plugins

sudo apt-get install k3b libk3b2-mp3 k3b-i18n

sudo apt-get install normalize-audio toolame movixmaker-2

sudo apt-get install vcdimager vorbis-tools sox transcode

sudo ln -s /usr/bin/normalize-audio /usr/bin/normalize

Renomeando nomes de arquivos com diferentes codificações

apt-get install convmv

convmv -f iso-8859-1 -t UTF-8 ação.txt --notest

Renomeará para aÃ§Ã£o.txt

CODECS extras para o Ubuntu

Instalar todos os pacotes sugeridos em:

https://wiki.ubuntu.com/RestrictedFormats?highlight=%28codecs%29%7C%28format%29

sudo apt-get install wget

wget -c http://www.debian-multimedia.org/pool/main/w/w32codecs/w32codecs_20061022-0.0_i386.deb

sudo dpkg -i w32codecs/w32codecs_20061022-0.0_i386.deb

sudo apt-get install gstreamer0.10-pitfdll gstreamer0.10-ffmpeg gstreamer0.10-plugins-bad gstreamer0.10-plugins-bad-multiverse gstreamer0.10-plugins-ugly gstreamer0.10-plugins-ugly-multiverse gxine libxine-main1 libxine-extracodecs ogle ogle-gui

Tocando MP3 no Ubuntu (e outras distros) - da Dicas-L Colaboração: Henrique Meira

O Ubuntu (e outras distros) por padrão não toca som MP3, para resolver isto basta instalar um plugin do GStreamer do conjunto "ugly".

No Ubuntu, você pode abrir o Gerenciador de Pacotes Synaptic e tentar localizar o pacote 'gstreamer0.10-plugins-ugly'.

Ou via linha de comando:

apt-get install gstreamer0.10-plugins-ugly

Este pacote está na seção Universe.

E isso já resolve tudo, já pode abrir o Rhythmbox por exemplo e ouvir suas músicas.

Criar imagem ISO de arquivos ou diretório

mkisofs -r -o nomedaimagem.iso /caminho/diretorioorigem/

Gravação de CD (ISO) via linha de comando

cdrecord dev=/dev/cdrw -v -eject nomeimagem.iso

Criando Imagem ISO via linha de comando

dd if=/dev/cdrw of=nomeimagem.iso

Montar Imagem ISO em partição

mount -o loop nomeimagem.iso /dev/hda3

Renomeando Múltiplos Arquivos via Shell

Colaboração: Euriam Barros de Araujo

Em algumas ocasiões, necessitamos renomear vários arquivos que possuem um determinado padrão.

Para facilitar este trabalho, segue o script abaixo:

```
for i in `ls files`
do
 x=${i#*left_padrao}
 y=${x%right_padrao*}
 z=${i%file_name*}
 mv $i $y-$z.extensao
done
```

Variáveis:

```
  * x: armazena a partir do padrão, removendo tudo à esquerda do mesmo
  * y: armazena o padrão, removendo tudo à direta do mesmo contido em x
  * z: armazena o nome do arquivo, exceto o padrão e sua extensão

    ==Exemplo:==
```

No diretório atual existem os arquivos abaixo e se deseja deslocar o tipo do trabalho para o início do arquivo:

```
trabalho-A.sxw

trabalho-B.sxw

trabalho-C.sxw

trabalho-D.sxw

for i in `ls trabalho*`
do
 x=${i#*-}
 y=${x%.*}
 z=${i%-[A-D]*}
 mv $i $y-$z.sxw
done
```

Renomeando Múltiplos Arquivos via Batch do DOS/Windows:

ren *.html *.php

Codificação de caracteres no Ubuntu Edgy:

Pesquisando na internet, reuni algumas dicas as quais funcionaram bem no meu computador.

1. Instalar locale com encoding pt_BR.ISO-8859-1:

sudo /usr/share/locales/install-language-pack pt_BR

2. Alterar o arquivo /etc/environment para:

LANG="pt_BR.ISO-8859-1"

3. Reconfigurar locales:

sudo dpkg-reconfigure locales

Lembrar de reiniciar o X para que as alterações tenham efeito.

Após restartar acessar um terminal e digitar:

locale

Para verificar como estão.

Julio Neto

Codificação de caracteres no Ubuntu 6.06

sudo /usr/share/locales/install-language-pack pt_BR

Instalar o arquivo localeconf:

sudo apt-get install localeconf

Rodar:

sudo dpkg-reconfigure localeconf

sim

ok

pt_BR ISO-8859-1

OK

Marcar todos

Em todos os próximos selecionar pt_BR ISO-8859-1

Restartar o ambiente gráfico.

Após restartar acessar um terminal e digitar:

locale

Para verificar como estão.

Julio Neto

Execução periódica de programa em linha de comando usando comando 'watch'

Caso você queira listar continuamente o conteúdo de um diretório para sua monitoração, basta digitar:

```
% watch -n 1 'ls -l'
```

Com a opção '-n' você pode especificar de quanto em quanto tempo o comando será executado pelo 'watch'.

Ao executar o comando acima, ele mostrará no topo da tela o comando que está em execução, data,

e horário em tempo real, bem como a saída do comando logo abaixo.

Outro exemplo útil:

Mostrar os usuários logados no samba em tempo real:

```
% watch -n 1 'smbstatus -b'
```

Seleção de Faixa Vertical de Texto com vim ou gvim

Um recurso muito útil e raro nos editores de texto e processadores inclusive.

Exemplo de uso: baixei um arquivo cujas linhas são numeradas e gostaria de remover somente os números das linhas.

- Abra o arquivo no vim ou gvim

- Posicione o cursor no ínicio da faixa

- Tecle Ctrl+v

- Mova o cursor para o final da faixa

- Tecle Ctrl+v novamente

- Com o cursor para baixo mova até onde deseja selecionar

Listando 10 maiores diretórios por ordem decrescente

du -s * | sort -rn | head -10

O "du -s *" cria uma lista com os tamanhos e nomes dos objetos no diretório corrente - no caso de diretórios o tamanho dos objetos dentro dele serão somados (parametro "-s") , o "sort -rn" pega a lista gerada pelo du e ordena a primeira coluna na forma numérica (o "-n") e de forma inversa (o parametro "-r") e o "head -10" mostra somente as 10 primeiras linhas.

Assim se você vai caçar comedores de disco em uma conta, talvez seja melhor utilizar a construção (inclusive ocultos)

du -s `ls -A` | sort -rn | head -10

Nessa linha de trocar a expansão por uma lista, o céu torna-se o limite

Rodando Java 5

Instalar pelo Synaptic o Java JDK 5 da Sun

Procurar por sun e marcar sun-java5-jdk

Após isso exportar a variável de ambiente com:

export JAVA_HOME=/usr/lib/jvm/java-1.5.0-sun-1.5.0.06/

Que versão do Java Utilizar? Execute e selecione a versão desejada:

sudo update-alternatives --config java

Uso do RAR na linha de comando do Linux:

rar a destino.rar /origem/* -m0 -v1495m

onde: 0 é o nível de compactação (0 - 5)

1495 é o tamanho dos arquivos em MBytes

Dica de: Ricardo de Miranda Amorim - Engenheiro de Computação

Teclas de Atalho no Ubuntu

- Criar lançador para gconf-editor
- Executar
- Expandir apps
- Rolar e expandir metacity
- Clicar em global_keybindings à esquerda
- À direita clicar em run_command_1 e na mesma linha abaixo de Valor digitar a

```
combinação de teclas para o programa a abrir, como por exemplo <Control><Alt>k
```

- Clicar em keybinding_commands e à direita digitar o nome do comando (se no path)

```
ou o caminho completo (exemplo kolourpaint).
```

Editor de texto amigável para o prompt

nano

ou

mcedit (do pacote mc: sudo apt-get install mc)

Adicionar Senha de Root

sudo passwd root

Entrar com a senha de usuário (administrador)

Depois entrar com a senha de root e repetir.

Usar root temporariamente

sudo -s

Configurar Problemas de Acentuação

(somente para a versão 5.10):

Abrir o terminal, maximizar a tela e executar:

sudo dpkg-reconfigure locales

Desmarcar todos os itens e marcar apenas "Pt_BR-ISO 8859-1"

Alterar IP pela console

/etc/network/interfaces

Após alterar:

ifdown eth0

ifup eth0

Localizar Arquivos Rapidamente

sudo updatedb (executar apenas quando houver instalado ou desinstalado algo)

locate nomearquivo

Outra alternativa

whereis nomecomando (só funciona para comandos executáveis do sistema operacional)

Bom Player de Filmes, MP3 e MP4

sudo apt-get install mplayer

Ativar DMA para o CDROM

hdparm -d1 /dev/cdrom

Para Instalar K3B dos Fontes

instalar xlibs-dev e kdebase-dev

apt-get install xlibs-dev

apt-get install kdebase-dev

Depois disso baixar os fontes, descompactar e compilar do site oficial.

Audotoria de Hardware

kinfocenter

sudo apt-get install hwinfo

Executar:

hwinfo --short > maquina.txt (Gera o arquivo com o hardware)

Outro bom:

sudo lshw > maquina.txt

Informações do hardware:

cat /proc/ide/hda/model

cat /proc/ide/hda/capacity

cat /proc/meminfo | grep MemTotal

cat /proc/cpuinfo

lspci
lspci -v
lspci -vv

ifconfig -a (dados da placa de rede)
netstat -na (quem está conectado nesta máquina)
netstat -nputa (portas e PID)
ntop -A (para dar uma senha)
ntop (informações da rede)
smbstatus - status do samba
dmesg |less

Listando Todas as Partições do Micro
sudo fdisk -l

Lista de Dispositivos PCIs
http://pciids.sourceforge.net/

- Guia Introdutório do Linux II

Tecla Windows no gnome e Outras teclas de atalho

http://www.dicas-l.com.br/print/20051125.html

Guia do ubuntu Linux

http://grubelilo.blogspot.com/2007/06/e-book-guia-do-ubuntu-linux.html

Configurando um servidor de email com Postfix e outros

- Configurando um servidor de email com Postfix, MySQL, SASL, MailScanner e MailWatch no CentOS

Dicas Sobre Pendrives, MP3, Câmeras e cia

Slax no Pendrive

http://www.vivaolinux.com.br/dicas/impressora.php?codigo=8383

Formatando e Consertando o Pendrive no Linux

http://br-net.org/post/formatando-o-pendrive-no-linux

Recuperando dados e fotos de câmeras digitais e pen drives

http://www.vivaolinux.com.br/dicas/verDica.php?codigo=8706 -

Formatar Pendrive

Verifique onde seu pendrive está montado inserindo-o na porta e executando o comando "df".

sudo umount /dev/sdb1

sudo mkdosfs -F 32 -n Tiago /dev/sdb1 (Tiago é o nome do volume)

Ou podemos usar o fdisk para remover e criar novamente a partição

Instalando o Slax no pendrive. Fácil, funcional e corrigido

http://www.vivaolinux.com.br/dicas/verDica.php?codigo=8383

Permissões - referência rápida

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=6420

Sistema de Vigilância Digital com ZoneMinder

http://www.guiadohardware.net/tutoriais/sistema-vigilancia-zoneminder/

Gravar CD de música pelo K3B

http://www.vivaolinux.com.br/dicas/impressora.php?codigo=8742

Evitando gargalos de hardware

http://canecas.blogspot.com/2007/06/sucata-iv-evitando-gargalos-de-hardware.html

VirtualBox Opção Livre para Virtualização

Tutorial de uso

http://localdomain.wordpress.com/2007/06/21/virtualbox/

Virtualizando com VirtualBox

Instalação e uso.

http://www.vivaolinux.com.br/artigos/impressora.php?codigo=6540

Rodando o Linux dentro do Windows "nativamente"

http://www.guiadohardware.net/noticias/2007-07/#4699e75c

Fazendo uma conexão bridge no VirtualBox

Agora para depois de ter instalado e querer interagir com o sistema operacional hospedeiro.

http://www.vivaolinux.com.br/artigos/impressora.php?codigo=6774

Personalizando a Tela de Login do Ubuntu 7.04

http://www.vivaolinux.com.br/dicas/impressora.php?codigo=8822

Instalando sem complicações o Java no Ubuntu 6.06, 6.10 e 7.04

http://www.vivaolinux.com.br/artigos/impressora.php?codigo=5502

http://blogetilico.wordpress.com/2007/01/24/instalacao-e-configuracao-java-jdk-no-ubuntu-610/

http://www.herbetferreira.com/instalando-jdk-6-no-ubuntu-704-the-feisty-fawn/

Guia de Inicialização de serviços no Linux - parte I

http://blog.riopro.com.br/2007/06/28/inicializacao-de-servicos-no-linux-parte-i/

Multimídia

Assista vídeos com suas suas músicas preferidas

Boa dica do Albert Guedes

http://www.vivaolinux.com.br/dicas/impressora.php?codigo=8850

Dicas sobre Desempenho

P Q P, quem puxou o freio de mão?

Extenso artigo do Antonio Carlos Lemos Júnior no Viva o Linux

http://www.vivaolinux.com.br/artigos/impressora.php?codigo=6046

Dicas Diversas

Listando diretórios e arquivos

http://www.vivaolinux.com.br/dicas/impressora.php?codigo=8855

Shell Scripts

Programando em shell-script (por Hugo Cisneiros)

http://www.devin.com.br/eitch/shell_script/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/Linux"
Page categories: PHP

Aplicativos em PHP/Referências/HTML

De Wikibooks

< Aplicativos em PHP | Referências

Conteúdo

- 1 HTML na Wikipedia
- 2 Título - Este aparece como título no Navegador
- 3 Body - Entra as tags <body> e </body> aparece praticamente toda a área útil da página
- 4 H1 - Texto para Título de tamanho 1. Tem tamanhos até o 6 (H6)
- 5 Tag form - Formulários são muito utilizadados para recceber informações do visitante
    - 5.1 Radio - Observe a prática de usar o mesmo name para agrupar vários radios
    - 5.2 Campo Desabilitado e Somente Leitura
- 6 Formatação
- 7 Fontes
- 8 Links(âncoras)
- 9 Imagem
- 10 Incorporar Vídeo e Som
- 11 Títulos
- 12 Select Múltiplo
- 13 Upload
- 14 Linha Horizontal
- 15 Frames
- 16 Respeitando a posição de digitação
- 17 Ótimo Tutorial sobre HTML do Tiago Daniel de Souza no site Linha de Código
- 18 HTML Wizards

HTML na Wikipedia

http://pt.wikipedia.org/wiki/Html

HTML (acrônimo para a expressão inglesa HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada para produzir páginas na Web. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto do "casamento" dos padrões HyTime e SGML.

HyTime é um padrão para a representação estruturada de hipermédia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), conectados por hiper-ligações. O padrão é independente de outros padrões de processamento de texto em geral.

Título - Este aparece como título no Navegador

<HTML><HEAD><TITLE>Exemplo de Página HTML</TITLE></HEAD>

Body - Entra as tags <body> e </body> aparece praticamente toda a área útil da página

Terá texto verde, fundo branco e a imagem "background.gif" de fundo.

<body text="green" bgcolor="white" link="#FFFF00" vlink="#FF0000" background="http://www.seudominio.com/imagens/background.gif">

H1 - Texto para Título de tamanho 1. Tem tamanhos até o 6 (H6)

<H1 ALIGN="center">Exemplo de Página HTML</H1>

Tag form - Formulários são muito utilizadados para recceber informações do visitante

Veja que existem diversos controles nos forms: text, radio, select, textarea, etc.

```
<form method="POST" action="teste.php">
<input type="hidden" name="recipient" value>
Nome:<input type="text" name="nome" size="40" maxlength="40">
Senha:<input type="password" name="senha" size="10" maxlength="10">
```

Radio - Observe a prática de usar o mesmo name para agrupar vários radios

```
<input type="radio" name="estadocivil" value="solteiro">Solteiro
<input type="radio" name="estadocivil" value="casado">Casado
<input type="radio" name="estadocivil" value="desquitado">Desquitado
<input type="radio" name="estadocivil" value="naoinformado">Não informado

<input type="checkbox" name="personalidade" value="Hostil">Hostil
<input type="checkbox" name="maturidade" value="Adulto">Adulto

<textarea name="comentario" rows="4" cols="60" wrap="virtual" style="color:green;background-color:yellow">Texto default</textarea>
```

Campo Desabilitado e Somente Leitura

```
<input type ="text" value ="Valor" disabled readonly>

<select name=selecao>
<option name="leite" value=1 SELECTED>Leite</option>
<option name="cafe" value=2>Café</option>
<option name="cha" value=3>Chá</option>
</select>

<SELECT NAME="grupo">
<OPTION VALUE="1">Equipamento
<OPTION VALUE="2">Mão de Obra
<OPTION VALUE="3">Ferramentas
<OPTION VALUE="4" SELECTED>Material
<OPTION VALUE="5">Serviços Auxiliares
<OPTION VALUE="6">Transporte
</SELECT>

<input type="submit" name="Request" value="Enviar"><input type="reset" name="Clear" value="Limpar">
</form>
```

Formatação

```
<b>Negrito</b> <i>Itálico</i> <u>Sublinhado</u> <s>Riscado</s> <sup>Sobrescrito</sup> <sub>Subscrito</sub>
```

Negrito *Itálico* Sublinhado ~~Riscado~~ ^{Sobrescrito} $_{\text{Subscrito}}$

Fontes

```
<font COLOR="#00FF00">This text is Green</font>  <!-- ou "green" -->
<font face="arial,helvetica,courier" size=1>Múltiplas Font Faces</font>
```

This text is Green

Múltiplas Font Faces

Links(âncoras)

<a HREF="http://www.2kweb.net" target="_BLANK">This links to 2K Communications</a>
<a HREF="mailto:sales@2kweb.net">Click here to e-mail our Sales Department</a>

Links para a página atual
1 - Criar destino - <a NAME="destino1"></a>
2 - Criar link - <a HREF="http://yourdomain.com/index.html#destino1">Click here to go there</a>

Imagem

<img src="http://www.2kweb.net/image-file.gif" alt="Explicação sobre a imagem" TITLE="Explicação sobre a imagem" ALIGN="LEFT" height=50 width=50>

Incorporar Vídeo e Som

<bgsound src="http://www.2kweb.net/sound.mid" loop=0"> (IE em loop. loop = -1 para parar)

<embed src="http://www.2kweb.net/sound.mid">

Títulos

```
<h1>Heading Size 1 - Maior</h1>
<h2>Heading Size 2 - Tamanho 2</h2>
<h3>Heading Size 3 - Tamanho 3</h3>
<h4>Heading Size 4 - Tamanho 4</h4>
<h5>Heading Size 5 - Tamanho 5</h5>
<h6>Heading Size 6 - Menor</h6>
```

Tabelas

```
<table border="1" width="630">
<tr>
        <td width="630" colspan=2> Largura de 630 pixels </td>
</tr>
<tr>
    <td width="550">Largura de 550 pixels</td> <td width="80">Largura de 80
pixels</td>
</tr>
</table>
```

Largura de 630 pixels

Largura de 550 pixels	Largura de 80 pixels

colspan = funde colunas

rowspan = funde linhas

```
<TABLE BORDER=1>
<TR><TH COLSPAN=2>Fusão de 2 colunas</TH></TR>
<TR><TD>linha1, coluna 1</TD><TD> linha 1, coluna 2</TD></TR>
<TR><TD>linha 2, coluna 1</TD><TD>linha 2, coluna 2</TD></TR>
<TR><TH ROWSPAN=3>Fusão de 3 linhas</TH><TD>uma linha</TD></TR>
<TR><TD>duas linhas</TD></TR>
<TR><TD>tres linhas</TD></TR>
</TABLE>
```

Fusão de Colunas	
linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2
Fusão de 3 linhas	uma linha
	duas linhas
	tres linhas

Select Múltiplo

```
<select name="combomulti" size="4" multiple>
<option value="lista1">Seleção 1</option>
<option value="lista2">Seleção 2</option>
<option value="lista3">Seleção 3</option>
<option value="lista4">Seleção 4</option>
<option value="lista5" selected>Seleção 5</option> <!--dEFAULT-->
<option value="lista6">Seleção 6</option>
</select>
```

Pressiona-se Ctrl ou Shift enquanto seclica ou move as setas para selecionar. </BODY> </HTML>

Upload

```
<FORM NAME="form1">
Arquivo a enviar: <INPUT TYPE="file" NAME="myUploadObject">
<P>Receber as propriedades<BR>
<INPUT TYPE="button" VALUE="Nome"   onClick="alert('name: ' +
document.form1.myUploadObject.name)">
<INPUT TYPE="button" VALUE="Valor"   onClick="alert('value: ' +
document.form1.myUploadObject.value)">
<BR></FORM>
```

Linha Horizontal

```
<HR width="40%" align="center">
```

Frames

Criar arquivo index.html com:

```
<frameset rows="25%,75%" frameborder="1" framespacing="0">
        <frame src=superior.html name=superior>
        <frameset cols=50%,50%>
                <frame src=esquerda.html name=esquerda frameborder="1"
framespacing="0" scrolling="yes">
                <frame src=direita.html name=direita frameborder="1"
framespacing="0" >
        </frameset>
</frameset><br>
```

Arquivo esquerda.html:

```
<a href="conteudohist.html" target="direita">História</a><br>
<a href="conteudomat.html" target="direita">Matemática</a><br>
<a href="conteudoport.html" target="direita">Português</a><br>
```

Criar os três arquivos acima (conteudohist.html, conteudomat.html e conteudoport.html).

Veja também um pequeno assistente de frames em:

http://ribafs.net/down/tut/html/wizards/

Respeitando a posição de digitação

Para que o navegador respeite todo o texto da mesma forma que digitamos devemos utilizar as tags

<pre> ... </pre>

<pre>

```
     Este texto
```

apareçerá nesta posição. </pre>

Ótimo Tutorial sobre HTML do Tiago Daniel de Souza no site Linha de Código

Guia Prático de HTML - Parte 1

Guia Prático de HTML - Parte 2

Guia Prático de HTML - Parte 3

Guia Prático de HTML - Parte 4

Guia Prático de HTML - Parte 5

HTML Wizards

http://ribafs.net/down/tut/html/wizards/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/HTML"
Page categories: PHP

Aplicativos em PHP/Referências/JavaScript

Conteúdo

- 1 JavaScript na Wikipedia
- 2 Introdução
- 3 Sintaxe
- 4 Primeiro script JavaScript
- 5 Location - onde usar JS?
- 6 Exemplo no head
- 7 Externo - Importando um arquivo JS externo
- 8 Operadores
  - 8.1 Aritméticos
  - 8.2 Exemplo de Operadores Javascript com Variáveis
  - 8.3 Operadores de Comparação
- 9 Variáveis
- 10 Nomes de variáveis
- 11 Funções
- 12 Eventos
- 13 Tipos de Instruções (Estruturas de Controle)
- 14 if
- 15 else
- 16 else if
- 17 for
- 18 Comentários
- 19 Array
- 20 Ordenação de arrays
- 21 Alert
- 22 Confirm
- 23 Prompt
- 24 Print
- 25 Time Delay
- 26 Preparar para redirecionar
- 27 PopUp
- 28 Date
- 29 Current Time Clock
- 30 Form
- 31 Validação de Formulários
- 32 Form
- 33 Void 0
- 34 Compatibilidade do JavaScript dos vários Navegadores
  - 34.1 Função para validar caracteres ao digitar compatível com IE e no FF
  - 34.2 Outra função para validar caracteres ao digitar (do mesmo site acima)
- 35 Comparando JavaScript com seu primo ActionScript (script do Flash)
- 36 JavaScript sem mistérios para Mozilla/Netscape e Microsoft Internet Explorer
- 37 Grandes repositórios de JavaScript
- 38 Referências

JavaScript na Wikipedia

http://pt.wikipedia.org/wiki/Javascript

JavaScript é uma linguagem de programação criada pela Netscape em 1995, que a princípio se chamava LiveScript, para atender, principalmente, as seguintes necessidades:

- Validação de formulários no lado cliente (programa navegador); - Interação com a página. Assim, foi feita como uma linguagem de script. Javascript tem sintaxe semelhante a do Java, mas é totalmente diferente no conceito e no uso.

1. Oferece tipagem dinâmica - tipos de variáveis não são definidos; 2. É interpretada, ao invés de compilada; 3. Possui ótimas ferramentas padrão para listagens (como as linguagens de script, de modo geral); 4. Oferece bom suporte a expressões regulares (característica também comum a linguagens de script).

Sua união com o CSS é conhecida como DHTML. Usando o Javascript, é possível modificar dinamicamente os estilos dos elementos da página em HTML.

Dada sua enorme versatilidade e utilidade ao lidar com ambientes em árvore (como um documento HTML), foi criado a partir desta linguagem um padrão ECMA, o ECMA-262, também conhecido como ECMAScript. Este padrão é seguido, por exemplo, pela linguagem ActionScript da Adobe(Antigamente Macromedia, porém a empresa foi vendida a Adobe).

Além de uso em navegadores processando páginas HTML dinâmicas, o JavaScript é hoje usado também na construção do navegador Mozilla, o qual oferece para a criação de sistemas GUI todo um conjunto de ferramentas (em sua versão normal como navegador, sem a necessidade de nenhum software adicional), que incluem (e não apenas) um interpretador de Javascript, um comunicador Javascript <-> C++ e um interpretador de XUL, linguagem criada para definir a interface gráfica de aplicações.

Introdução

O JavaScript é uma ferramenta muito util para a construção de Web Sites, especialmente se utilizada em conjunto com HTML, CSS, PHP, entre outras ferramentas.

Também pode criar diversas ferramentas para sites:

```
* Jogos
* Relógios
* Animações com mouse
* Menus com select
* Mensagens de alerta
* Janelas Popup
* Validação de Formulários HTML
```

Sintaxe

Como escrever código JavaScript?

```
* Usar a tag <script para dizer ao browse que você está usando Javascript.
* Escrever ou fazer download de Javascript
* Testar scripts
```

Primeiro script JavaScript

HTML e JS

```
<html>
<body>
<script type="text/javascript">
<!--
document.write("Olá Mundo do JavaScript!")
//-->
</script>
</body>
</html>
```

Location - onde usar JS?

```
  # Dentro da tag head
  # Na tag body
  # Em um arquivo externo
```

Exemplo no head

```
<html>
<head>
<script type="text/javascript">
<!--
function popup() {
alert("Olá Mundo do JavaScript (function)")
}
//-->
</script>
</head>
<body>
<input type="button" onclick="popup()" value="Executar">
</body>
</html>
```

Externo - Importando um arquivo JS externo

O arquivo deve estar em JS e somente com sintaxe JS e usar extensão .js

Importante: os arquivos a serem incluídos não devem conter as tags

<script> e </script>.

```
<html>
```

```
<head>
<script src="meujs.js">
</script>
</head>
<body>
<input type="button" onclick="popup()" value="Clique Me!">
</body>
</html>
```

Operadores

Aritméticos

```
Operator Descrição     Example
+              Addition          2 + 4
-                Subtraction     6 - 2
*               Multiplication  5 * 3
/                Division           15 / 3
%              Modulus          43 % 10
```

Exemplo de Operadores Javascript com Variáveis

```
<body>
<script type="text/javascript">
<!--
var two = 2
var ten = 10
var linebreak = "<br />"

document.write("two plus ten = ")
result = two + ten
document.write(result)
document.write(linebreak)


document.write("ten * ten = ")
result = ten * ten
document.write(result)
document.write(linebreak)

document.write("ten / two = ")
result = ten / two
document.write(result)
//-->
</script>
</body>
```

Operadores de Comparação

```
Operator    Descrição         Example     Result
```

```
==              Equal To          $x == $y     false
!=              Not Equal To      $x != $y     true
<              Less Than          $x < $y     true
>              Greater Than     $x > $y     false
<=              Menor ou igual      $x <= $y     true
>=              Maior ou igual      $x >= $y     false
```

Variáveis

Podemos imaginar uma variável como uma sexta de frutas e os dados das variáveis como as frutas da sexta.

Quando usar uma variável pela primeira vez é uma boa prática de programação usar a palavra reservada var antes do nome da variável.

```
<body>
<script type="text/javascript">
<!--
var linebreak = "<br />"
var my_var = "Hello World!"

document.write(my_var)
document.write(linebreak)

my_var = "I am learning javascript!"
document.write(my_var)
document.write(linebreak)

my_var = "Script is Finishing up..."
document.write(my_var)
//-->
</script>
</body>
```

Nomes de variáveis

- evitar palavras reservadas do JavaScript

- escolher nomes descritivos

- nomes de variáveis não podem iniciar com algarismos

- uma boa prática é iniciar com minúsculas

Funções

Funções em JavaScript são similares a funções em outras linguagens como PHP, C, C++, Java.

Uma função é um pedaço de código que fica dormindo até que seja chamado para "funcionar".

Ao invés de digitar todo o código sempre que precisar, podemos simplesmente chamar novamente a função. Isto é conhecido como reutilização de código.

```
<html>
<head>
<script type="text/javascript">
<!--
function popup() {
    alert("Olá Mundo do JS!")
}
//-->
</script>
</head>
<body>
<input type="button" onclick="popup()" value="Executar">
</body>
</html>
```

Eventos

Este é o grande recurso do JavaScript, que possibilita interagir com o usuário no site, como:

- clique do mouse

- página sendo carregada

- mouse movendo sobre algum componente da página

- Seleção de item de Select

- Toque de tecla

- etc

```
<html>
<head>
<script type="text/javascript">
<!--
function popup() {
alert("Hello World")
//-->
</script>
</head>
<body>
<input type="button" value="Click Me!" onclick="popup()"><br />
<a href="#" onmouseover="" onMouseout="popup()"">
Hover Me!</a>
</body>
</html>
```

Todo código em JavaScript é formado por instruções (statments).

Uma instrução é uma variável igual a um valor.

Como também é algo como uma chamada de função, por exemplo document.write(). As instruções definem o que o script deve fazer e como fazer.

Em algumas linguagens como o PHP, o final das instruções obrigatoriamente deve terminar com ponto e vírgula. Mas em JavaScript o ; ao final das instruções é opcional. Obrigatoriamente as

instruções terminam com uma quebra de linha.

Tipos de Instruções (Estruturas de Controle)

- Condicionais
- Loops
- Manipulação de Objetos
- Comentários
- Manupulação de exceções

if

```
<script type="text/javascript">
<!--
var myNum = 7;

if(myNum == 7){
    document.write("Lucky 7!");
}
//-->
</script>
```

else

```
<script type="text/javascript">
<!--
var myNum = 10;

if(myNum == 7){
    document.write("Lucky 7!");
}else{
    document.write("You're not very lucky today...");
}
//-->
</script>
```

else if

```
<script type="text/javascript">
<!--
var visitor = "principal";

if(visitor == "teacher"){
    document.write("My dog ate my homework...");
}else if(visitor == "principal"){
```

```
    document.write("What stink bombs?");
} else {
    document.write("How do you do?");
}
//-->
</script>
```

while

```
<script type="text/javascript">
<!--
var myCounter = 0;
var linebreak = "<br />";
document.write("While loop is beginning");
document.write(linebreak);

while(myCounter < 10){
    document.write("myCounter = " + myCounter);
    document.write(linebreak);
    myCounter++;
}

document.write("While loop is finished!");
</script>
```

for

```
<script type="text/javascript">
<!--
var linebreak = "<br />";
document.write("For loop code is beginning");
document.write(linebreak);

for(i = 0; i < 5; i++){
    document.write("Counter i = " + i);
    document.write(linebreak);
}

document.write("For loop code is finished!");
</script>
```

Comentários

```
<script type="text/javascript">
<!--
// This is a single line Javascript comment

document.write("I have comments in my Javascript code!");
//document.write("You can't see this!");
//-->
```

```
</script>
```

Array

São tipos especiais de variáveis.

Valores são armazenados em arrays passando o nome do array e a localização no array entre colchetes.

myArray[2] = "Hello World";

Valores em arrays são acessados com o nome do array e a localização do valor:

myArray[2]

JavaScript traz algumas funções para trabalhar com arrays.

```
<script type="text/javascript">
<!--
var myArray = new Array();

myArray[0] = "Baseball";
myArray[1] = "Cricket";
myArray[2] = "Football";

document.write(myArray[0] + myArray[1] + myArray[2]);
//-->
</script>
```

Ordenação de arrays

```
<script type="text/javascript">
<!--
var myArray = new Array();

myArray[0] = "Baseball";
myArray[1] = "Cricket";
myArray[2] = "Football";

myArray.sort();

document.write(myArray[0] + myArray[1] + myArray[2]);
//-->
</script>
```

Alert

Alert emite uma caixa de mensagem (janela popup) com mensagem texto.

```
<form>
<input type="button" onclick= "alert('Realmente excluir o registro atual?')"
value="Corfirmar Exclusão">
</form>
```

Confirm

Solicita confirmação com uma mensagem e exibindo dois botões para confirmar ou desistir.

```
<html>
<head>
<script type="text/javascript">
<!--
function confirmation() {
    var answer = confirm("Deixar o curso_php.org?")
    if (answer){
        alert("Até outro dia!")
        window.location = "http://www.google.com/";
    }
    else{
        alert("Grato por permanecer!")
    }
}
//-->
</script>
</head>
<body>
<form>
<input type="button" onclick="confirmation()" value="Deixar o curso_php.org">
</form>
</body>
</html>
```

Prompt

Abra uma janela com uma mensagem e uma caixa de texto para receber resposta do usuário.

```
<head>
<script type="text/javascript">
<!--
function prompter() {
var reply = prompt("Olá amigo visitante! Qual seu nome?", "")
alert ('Seu nome é ' + reply)
}
//-->
</script>
</head>
<body>
```

```
<input type="button" onclick="prompter()" value="Dizer meu nome!">
</body>
```

Print

Abre o diálogo de impressão do browser para imprimir a janela atual

```
<form>
<input type="button" value="Print This Page" onClick="window.print()" />
</form>
```

Redirect

```
<script type="text/javascript">
<!--
window.location = "http://www.google.com/"
//-->
</script>
```

Time Delay

```
<html>
<head>
<script type="text/javascript">
<!--
function delayer(){
    window.location = "../javascriptredirect.php"
}
//-->
</script>
</head>
<body onLoad="setTimeout('delayer()', 5000)">
```

Preparar para redirecionar

This page is a time delay redirect, please update your bookmarks to our new location!

</body> </html>

PopUp

<head> <script type="text/javascript"> </script> </head> <body> <form> <input type="button" onClick="myPopup()" value="POP!"> </form>

CLICK ME TOO!

</body>

```
   *  dependent - Subwindow closes if parent(the window that opened it) window
closes
   * fullscreen - Display browser in full screen mode
   * height - The height of the new window, in pixels
   * width - The width of the new window, in pixels
   * left - Pixel offset from the left side of the screen
   * top - Pixel offset from the top of the screen
   * resizable - Allow the user to resize the window or prevent resizing
   * status - Display the status bar or not
```

<head> <script type="text/javascript"> </script> </head> <body> <form> <input type="button" onClick="myPopup2()" value="POP2!"> </form>

CLICK ME TOO!

</body>

Date

Isto é agora

```
<script type="text/javascript">
<!--
var currentTime = new Date()
//-->
</script>

   * getTime() - Number of milliseconds since 1/1/1970 @ 12:00 AM
   * getSeconds() - Number of seconds (0-59)
   * getMinutes() - Number of minutes (0-59)
   * getHours() - Number of hours (0-23)
   * getDay() - Day of the week(0-6). 0 = Sunday, ... , 6 = Saturday
   * getDate() - Day of the month (0-31)
   * getMonth() - Number of month (0-11)
   * getFullYear() - The four digit year (1970-9999)
```

Agora podemos imprimir as informações. Usaremos os métodos getDate, getMonth, e getFullYear neste exemplo.

Agora

```
<script type="text/javascript">
<!--
var currentTime = new Date()
var month = currentTime.getMonth() + 1
var day = currentTime.getDate()
```

```
var year = currentTime.getFullYear()
document.write(month + "/" + day + "/" + year)
//-->
</script>
```

Current Time Clock

Agora

```
<script type="text/javascript">
<!--
var currentTime = new Date()
var hours = currentTime.getHours()
var minutes = currentTime.getMinutes()
if (minutes < 10)
minutes = "0" + minutes
document.write(hours + ":" + minutes + " ")
if(hours > 11){
document.write("PM")
} else {
document.write("AM")
}
//-->
</script>
```

Form

Checando por não vazio

function notEmpty(elem){ var str = elem.value; if(str.length == 0){

```
   alert("You must fill in all required fields (*)");
   return false;} else {
   return true;}
```

}

Validação de Formulários

function formValidation(form){

```
   if(notEmpty(form.req1)){
       if(notEmpty(form.req2)){
           return true;
       }
   } else {
       return false;
   }
```

}

Form

<head> <script type="text/javascript"> </script> </head> <body> <form name="example" onSubmit="return formValidation(this)">

- Required 1:<input type="text" name="req1" />
- Required 2:<input type="text" name="req2" />

<input type="submit" />

</form> </body>

Void 0

<a href="javascript: alert('News Flash!')">News Flash</a>

<a href="javascript: void(0)">I am a useless link</a>

<a href="javascript: void(myNum=10);alert('myNum = '+myNum)"> Set myNum Please</a>

Compatibilidade do JavaScript dos vários Navegadores

Função para validar caracteres ao digitar compatível com IE e no FF

```
<script>
function onKeyPressNaoAlgarismos(e)
{
        var key = window.event ? e.keyCode : e.which;
        var keychar = String.fromCharCode(key);
        reg = /\d/;
        return !reg.test(keychar);
}

function onKeyPressSoAlgarismos(e)
{
        var key = window.event ? e.keyCode : e.which;
        var keychar = String.fromCharCode(key);
        reg = /\d/;
        return reg.test(keychar);
}
</script>
```

Testar:

```
<form>
Permite só de 0 - 9: <input name="algarismos" type="text" onkeypress="return
onKeyPressNaoAlgarismos(event);" /><br>
Impede de 0 - 9: <input name="nao_alg" type="text" onkeypress="return
```

```
onKeyPressSoAlgarismos(event);" /><br>
</form>
```

Adaptado de Tutorials - KeyPress validation http://www.mredkj.com/tutorials/validate.html

Outra função para validar caracteres ao digitar (do mesmo site acima)

function blockNonNumbers(obj, e, allowDecimal, allowNegative) { var key; var isCtrl = false; var keychar; var reg;

if(window.event) { key = e.keyCode; isCtrl = window.event.ctrlKey } else if(e.which) { key = e.which; isCtrl = e.ctrlKey; }

if (isNaN(key)) return true;

keychar = String.fromCharCode(key);

// check for backspace or delete, or if Ctrl was pressed if (key == 8 || isCtrl) { return true; }

reg = /\d/; var isFirstN = allowNegative ? keychar == '-' && obj.value.indexOf('-') == -1 : false; var isFirstD = allowDecimal ? keychar == '.' && obj.value.indexOf('.') == -1 : false;

return isFirstN || isFirstD || reg.test(keychar); }

Comparando JavaScript com seu primo ActionScript (script do Flash)

http://www.webreference.com/programming/javascript/j_s/column2/

JavaScript sem mistérios para Mozilla/Netscape e Microsoft Internet Explorer

http://www.vivaolinux.com.br/dicas/impressora.php?codigo=8872

Grandes repositórios de JavaScript

http://www.javafile.com/

http://www.java-scripts.net/

http://dhtml-menu.com/

http://www.tizag.com/javascriptT/ (site com o original deste tutorial)

http://javascript.internet.com/

Referências

http://developer.mozilla.org/en/docs/JavaScript

http://www.javascriptkit.com/jsref/

http://javascript-reference.info/

http://www.w3schools.com/jsref/default.asp

http://docs.sun.com/source/816-6408-10/

http://www.devguru.com/technologies/JavaScript/home.asp

http://wp.netscape.com/eng/mozilla/3.0/handbook/javascript/

http://devedge-temp.mozilla.org/library/manuals/2000/javascript/1.5/reference/

http://www.comptechdoc.org/independent/web/cgi/javamanual/

http://www.quirksmode.org/js/contents.html

http://argento.bu.edu/~ccruz/javascript/reference.html

http://www.java2s.com/Code/JavaScriptReference/CatalogJavaScriptReference.htm

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/JavaScript"
Page categories: PHP

Aplicativos em PHP/Referências/CSS

De Wikibooks

< Aplicativos em PHP | Referências

Conteúdo

- 1 CSS na Wikipedia
- 2 Agrupar Seletores
- 3 Cor do Texto
- 4 Cor de Fundo
- 5 Imagem de Fundo
- 6 Exemplo
- 7 Tipos de Fontes
- 8 Propriedades dos Estilos de Texto
- 9 Tipos de Folhas de Estilo
    - 9.1 - Integrada ao HTML
    - 9.2 - Embutida na página
    - 9.3 - Externo
- 10 Usando duas Folhas de Estilo
- 11 Folhas de Estilo Persistentes
- 12 Folhas de Estilo Default
- 13 Folhas de Estilo Alternativas
- 14 Compatibilidade entre Navegadores
- 15 Posicionando com Estilo
    - 15.1 Propiedades de Texto
    - 15.2 Propriedades de Fontes
    - 15.3 Propriedades de Quadro
    - 15.4 Propriedades de Cor e Fundo
    - 15.5 Propriedades de Classificação
- 16 Ferramentas para auxílio no CSS
- 17 Referências

CSS na Wikipedia

http://pt.wikipedia.org/wiki/Cascading_Style_Sheets

Cascading Style Sheets, ou simplesmente CSS, é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

Ao invés de colocar a formatação dentro do documento, o desenvolvedor cria um link (ligação) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta portanto modificar apenas um arquivo.

Com a variação de atualizações dos navegadores (browsers) como Internet Explorer que ficou sem nova versão de 2001 a 2006, o suporte ao CSS pode variar. O Internet Explorer 6, por exemplo, tem suporte total a CSS1 e praticamente nulo a CSS2. Navegadores mais modernos como Opera, Internet Explorer 7 e Mozilla Firefox tem suporte maior, inclusive até a CSS 3, ainda em desenvolvimento.

A interpretação dos browsers pode ser avaliada com o teste Acid2, que se tornou uma forma base de revelar quão eficiente é o suporte de CSS, fazendo com que a nova versão em desenvolvimento do Firefox seja totalmente compatível a ele assim como o Opera já é.

Agrupar Seletores

Podemos agrupar vários seletores da seguinte forma:

```
h1,h2,h3,h4,h5,h6
{
color: green;
}
```

Cor do Texto

```
cortexto{color:green;}
```

Cor de Fundo

```
corfundo{background-color:blue;}
```

Imagem de Fundo

```
imagem{ background-image:url(fondo.jpg);}
```

Exemplo

```
<HTML><HEAD><TITLE>Exemplo</TITLE>
<STYLE type="text/css">
<!--
P{background-color:black;
 color:purple; width:400px;
 border-color:blue; border-width:12px;}
.tuestilo{background-image:url(greenmar.gif); color:yellow;
border-style:inset; border-width:5pt; float:left; width:60px;}
H1{background-color:#afaefd; width:40px; clear:both;
padding-right:10px; padding-top:12px; color:navy;}
H3{background-color:blue; margin-left:14px; margin-top:35px;}
-->
</STYLE></HEAD>
<BODY>
<P>Este parágrafo terá texto violeta com em fundo preto e largura de 400 pixels,
assim como uma borda azul com uma altura de 12 pixels. </P>
<DIV class="tuestilo">Este parágrafo usará a clase de estilo "tuestilo", usará
uma imagen com fundo assim como uma borda estilo "inset"
com uma altura de 5 pontos, permitirá que haja elementos flutuando à sua direita
e terá uma largura de 60 pixels.</DIV>
<H1>Entre outras características este bloco tem uma margen interna superior de
12 pixels e uma margen interna direita de 10 pixels.</H1>
<H3>Este bloco usará uma margen esquerda de 14 pixels e uma superior de 35
pixels</H3>
</DIV>
</BODY></HTML>
```

Tipos de Fontes

```
<HTML><HEAD><TITLE>UN SIMPLE EJEMPLO </TITLE>
<STYLE TYPE="text/css">
<!--
.meu_estilo{font-size:14pt; font-family:Comic Sans MS; font-weight:Bold; font-
style:italic;}
H3 {font-size:25px; font-family:Symbol; font-weight:bold;}
-->
</STYLE></HEAD>
<BODY>
<DIV class="miestilo">Estas linhas estão usando o estilo chamado
"meu_estilo".</DIV>
<H3>A TAG H3 foi redefinida</H3>
</BODY></HTML>
```

Propriedades dos Estilos de Texto

```
<HTML><HEAD><TITLE> MAIS EXEMPLOS </TITLE>
<STYLE type="text/css">
.subrayado{text-decoration:underline; color:red;}
.tachado{text-decoration:line-through; color:blue;}
.testado{text-decoration:overline; color:purple;}
.parpadea{text-decoration:blink; color:green;}
```

```
.sangria{text-indent:100px; color:navy;}
.centrado{text-align:center; color:olive;}
.capitales{text-transform:capitalize; color:silver;}
.mayusculas{text-transform:uppercase; color:#aace34;}
.demoaltura{line-height:150%; color:#eeee44;}
</STYLE>
</HEAD>
<BODY>
<SPAN class="subrayado">SUBRAYADO</SPAN> <SPAN class="tachado">TACHADO</SPAN>
<SPAN class="testado">OVERLINE</SPAN> <SPAN class="parpadea">Parpadea</span>
<BR><BR><DIV class="sangria">SANGRIA abcdefghijklmnñopqrstuvwxyz,
abcdefghijklmnñopqrstuvwxyz
abcdefghijklmnñopqrstuvwxyz, abcdefghijklmnñopqrstuvwxyz,
abcdefghijklmnñopqrstuvwxyz</DIV>
<br>
<DIV class="centrado">CENTRADO</Div>
<br>
<DIV class="capitales">capitales, abcdef, ghijkl, mnñopq, rstuvw, xyz,
capitales.</diV>
<br>
<DIV class="mayusculas">abcdefghijklmnñopqrstuvwxyz</DIV>
<p class="demoaltura">
AAAAAAAAAA<br>BBBBBBBB<br>CCCCCCCCC<br>DDDDDDDDD
</p>
</BODY></HTML>
```

Tipos de Folhas de Estilo

- Integrado

- Embutido

- Externo

- Integrada ao HTML

Usado nas próprias TAGs HTML, em seu atributo STYLE, como:

```
<P STYLE="color: #00ff00;">
```

Para evitar problemas de compatibilidade usar entre <head> e </head>:

<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">

- Embutida na página

Se usa no cabeçalho do HTML com as tags <STYLE> ... </STYLE>

```
<STYLE TYPE="text/css">
<!--
IMG { vertical-align: 50%;}
P {color: #00ff00;}
A {font-family: sans-serif;}
```

```
A:link {color: #0000ee;}
A:visited {color: #551a8b;}
A:active {color: #ff0000;}
-->
</STYLE>
```

As linhas:

```
<!--
e
-->
```

São utilizadas para evitar problemas em navegadores sem suporte a CSS, que no caso considerarão como comentário.

Cria-se trechos de código como o acima na TAG head e utiliza-se os elementos definidos nas TAGs HTML da página, assim:

```
<P>Este parágrafo terá a cor do texto
Azul, pois o elemento P foi definido no trecho acima
como azul </P>
```

- Externo

A opção de uso do CSS recomendada é a de criação de um arquivo externo inteiramente em CSS e se importando esse na página, assim:

```
<STYLE TYPE="text/css">
<!--
@import url(http://pagina.net)
-->
</STYLE>
```

Para compatibilidade com navegadores mais antigos usa-se a sinta-se:

<LINK REL="stylesheet" HREF="estilo.css" TYPE="text/css">

Usando duas Folhas de Estilo

(Exibir e Imprimir)

```
<LINK REL="stylesheet" MEDIA="screen" HREF="tela.css" TYPE="text/css">
<LINK REL="stylesheet" MEDIA="print" HREF="impressora.css" TYPE="text/css">
```

Quando o documento é exibido na tela se utilizará o "tela.css", ignorando la segunda folha. E ao imprimir a página se usará a folha "impressora.css", ignorando a anterior.

Folhas de Estilo Persistentes

(Que os navegadores não podem desabilitar)

Aplica-se sempre ao documento sem possibilidade de se desabilitá-la.

Folhas de Estilo Default

Semelhantes às anteriores mas contendo o atributo NAME. Na carga do navegador elas são lançadas na memória e utilizadas na página mas o navegador pode decidir anular seu efeito.

Folhas de Estilo Alternativas

Estas não se aplicam ao documento a menos que o navegador as selecione.

```
<LINK REL="stylesheet" HREF="folhapersistente.css" TYPE="text/css">
<LINK REL="stylesheet" NAME="Folha Default" HREF="default.css" TYPE="text/css">
<LINK REL="alternate stylesheet" NAME="Folha Alternativa" HREF="alternativa.css"
TYPE="text/css">
```

Compatibilidade entre Navegadores

Vale lembrar que tanto o CSS quanto o JavaScript e outros recursos utilizados na construção de sites não têm seu uso inteiramente padronizado pelos navegadores. Alguns navegadores não seguem ao pé da letra os padrões o que acarreta alguns sites não serem exibidos de forma adequada por todos os navegadores mas apenas por aquele para o qual o programador previu em sua programação. Cabe a nós desenvolvedores evitar o uso de práticas e recursos que não sejam compatíveis com os principais navegadores, para evitar que alguns visitantes tenham problemas ao acessar o site.

Posicionando com Estilo

```
<html><head><title>Posicionando com CSS</title></head>
<body>

<div id="listagem"
style="position:absolute;left:1%;top:1%;width:20%;height:95%;border: solid
brown">
<P align="center"><?php include("includes/index.php"); ?></P>
</div>

<div id="cabecalho"
style="position:absolute;left:22%;top:1%;width:60%;height:10%;border: solid
brown">
<P align="center"><?php include("includes/cabecalho.php"); ?></P>
```

```
</div>

<div id="fck"
style="position:absolute;left:22%;top:13%;width:60%;height:75%;border: solid
brown">
<P align="center"><?php include("includes/fck.php"); ?></P>
</div>

<div id="tags"
style="position:absolute;right:1%;top:1%;width:15.5%;height:95%;border: solid
brown">
<P align="center"><?php include("includes/tags.php"); ?></P>
</div>

<div id="rodape"
style="position:absolute;left:22%;top:90.3%;width:60%;height:5.6%;border: solid
brown">
<P align="center"><?php include("includes/rodape.php"); ?></P>
</div>
</body>
</html>
```

Algumas Propriedades

Propiedades de Texto

```
word-spacing
letter-spacing
text-decoration
vertical-align
text-transform
text-align
text-indent
line-height
```

Propriedades de Fontes

```
font-family
font-style
font-variant
font-weight
font-size
font
```

Propriedades de Quadro

```
margin-top
margin-botton
margin-right
margin-left
```

```
margin
padding-top
padding-right
padding-botton
padding-left
padding
border-top-width
border-right-width
border-botton-width
border-left-width
border-width
border-color
border-style
border-top
border-right
border-botton
border-left
border
width
height
float
clear
```

Propriedades de Cor e Fundo

```
color
background-color
background-image
background-repeat
background-attachment
background-position
background
```

Propriedades de Classificação

```
display
white-space
list-style-type
list-style-image
list-style-position
list-style
```

Ferramentas para auxílio no CSS

Existem diversas boas ferramentas para trabalhar ajudando no CSS de sites.

Dreamweaver

É uma ferramenta comercial somente for Windows muito utilizada, especialmente pelos designers.

Aptana

http://www.aptana.com, que inclusive oferece um plugin para o Eclipse. Ele mostra a compatibilidade de cada tag com os principais navegadores, além de contar com ótimo recurso de auto-completar o HTML, JavaScript e CSS. É uma ferramenta free for Windows, Linux e Mac.

BlueFish e Quanta

São muito boas ferramentas para Linux.

Referências

- Free Menu Designs (http://e-lusion.com/design/menu/)

- Gerando estilo CSS para botões - http://

- Dicas de CSS - http://www.phpriot.com/d/articles/client-side/css-tips/index.html

- 53 Técnicas de CSS que você não pode viver sem - http://www.smashingmagazine.com/2007/01/19/53-css-techniques-you-couldnt-live-without/

- Tutorial básico de hojas de estilo en cascada - http://www.geocities.com/lunamatic/tutcss/index.html

- Referências CSS 2.1 - http://www.w3.org/TR/2004/CR-CSS21-20040225

- Recomendações CSS-1 - http://www.w3.org/TR/REC-CSS1

- Recomendações CSS-2 - http://www.w3.org/TR/WD-css2

- CSS Wizard - http://ribafs.net/down/tut/html/wizards/csswiz.html

- CSS para Webdesign - http://www.maujor.com/

- glish.com: CSS Layout Techniques - http://www.glish.com/css/

- tanfa - CSS design, style & fun - http://www.tanfa.co.uk/

- CSS Zen Garden - http://www.csszengarden.com/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/CSS"
Page categories: PHP

Aplicativos em PHP/Referências/PHP/PHP1

De Wikibooks

Conteúdo

- 1 ALERTA - PHP4 chegando ao fim da linha
- 2 - INTRODUÇÃO
  - 2.1 Popularidade do PHP
  - 2.2 Estatísticas sobre o PHP
  - 2.3 História do PHP
  - 2.4 Delimitadores do PHP
  - 2.5 Grandes Forças do PHP
  - 2.6 O Que é Possível Realizar em PHP?
- 3 - TUTORIAL SIMPLES
  - 3.1 Informações sobre o PHP
  - 3.2 Checar navegador em uso
  - 3.3 Teste com echo e print
- 4 - CONFIGURAÇÕES
  - 4.1 php.ini
  - 4.2 httpd.conf
- 5 - Referência da Linguagem
  - 5.1 Separador de Instruções
  - 5.2 Comentários
- 6 - Tipos de Dados
  - 6.1 Tipo de Variável
  - 6.2 Funções que Retornal o Tipo
  - 6.3 Casting
  - 6.4 Boleanos
  - 6.5 Valores que são considerados FALSE
  - 6.6 Os demais são TRUE
  - 6.7 Inteiros
  - 6.8 Alerta
  - 6.9 Ponto Flutuante
  - 6.10 Strings
  - 6.11 Especificando Strings
  - 6.12 Nulos em PHP
- 7 - VARIÁVEIS
  - 7.1 Variáveis Predefinidas
  - 7.2 Escopo de Variáveis
  - 7.3 Escopo de variáveis em funções
  - 7.4 Usando $GLOBALS no lugar de global
  - 7.5 Utilizando Variáveis Estáticas
  - 7.6 Variáveis Variáveis
  - 7.7 Determiando o Tipo das Variáveis
- 8 - Constantes
  - 8.1 Constantes Mágicas
  - 8.2 Algumas constantes "mágicas" do PHP
- 9 - EXPRESSÕES
- 10 - Operadores
  - 10.1 Tipos de Operadores
  - 10.2 Precedência de Operadores
  - 10.3 Exemplo
  - 10.4 Para facilitar a percepção das precedências usa-se parêntesis
  - 10.5 Operadores Aritméticos
  - 10.6 Exemplos de operadores aritméticos
  - 10.7 Operadores de Atribuição
  - 10.8 Operadores de Controle de Erro
  - 10.9 Operadores de Execução
  - 10.10 Operadores de Incremento e de Decremento
  - 10.11 Operadores Lógicos
  - 10.12 Operadores de String
  - 10.13 Convertendo strings em números
  - 10.14 Operações com Strings
  - $10.15 Operadores com Arrays

ALERTA - PHP4 chegando ao fim da linha

Alerta para que quem ainda não migrou para o PHP 5 inicie sua migração:

Depois de 3 anos de PHP5 e com o PHP 6 chegando, a versão 4 será descontinuada a partir do ano que vem. Bugs críticos ainda serão corridos até o final de agosto de 2008.

Enviado por Marcos Alexandre Lemos Rodrigues para o br-linux (marcosalexandre·rodriguesΘgmail·com)

Notícia original em ingles:

PHP 4 end of life announcement [13-Jul-2007]

Today it is exactly three years ago since PHP 5 has been released. In those three years it has seen many improvements over PHP 4. PHP 5 is fast, stable & production-ready and as PHP 6 is on the way, PHP 4 will be discontinued.

The PHP development team hereby announces that support for PHP 4 will continue until the end of this year only. After 2007-12-31 there will be no more releases of PHP 4.4. We will continue to make critical security fixes available on a case-by-case basis until 2008-08-08. Please use the rest of this year to make your application suitable to run on PHP 5.

For documentation on migration for PHP 4 to PHP 5, we would like to point you to our migration guide. There is additional information available in the PHP 5.0 to PHP 5.1 and PHP 5.1 to PHP 5.2 migration guides as well.

Documentos para ajudar na migração:

Migration Guide - http://www.php.net/manual/en/migration5.php Também - http://www.php.net/manual/en/migration51.php e http://www.php.net/manual/en/migration52.php

Este texto teve como principal fonte de informações o manual oficial do PHP em http://www.php.net/manual/pt_BR/

- INTRODUÇÃO

O Que é PHP?

É uma linguagem de script, Open Source, de uso geral, no lado do servidor, embutível no HTML e especialmente voltada para o desenvolvimento Web. Originalmente chamou-se PHP (Personal Home Page) mas depois teve seu nome alterado pela comunidade para PHP: Hypertext Processor, um acrônimo recursivo.

Um pequneno script em PHP

```
<?php
echo "Script em PHP!";
?>
```

Popularidade do PHP

O PHP atualmente deve ser a linguagem web mais usada no planeta.

Segundo estatísticas do site:

http://www.drews.cx/2006/03/23/php-usage-stats-go-up-again/

Onde cita as 4 mais populares linguagens: C, C++, Java e PHP. Nesta ordem.

Observação: todas oriundas do C.

Na prática, entre as linguagens Web o PHP fica em segundo lugar. Acontece que Java é não somente uma linguagem mas uma plataforma, contando com recursos para desktop, microedition, etc. Já PHP tem o uso restrito praticamente à Web.

Veja a lista geral:

http://www.tiobe.com/index.htm?tiobe_index

Em março/2006 ele já está presente em mais de 20 milhões de domínios ao redor do planeta (http://www.php.net/usage.php).

Estatísticas sobre o PHP

http://www.nexen.net/chiffres_cles/phpversion/php_statistics_for_april_2006.php

História do PHP

- Criado em 1995 por Rasmus Lerdorf, na forma de scripts Perl para coletar estatísticas online de seu currículo. Com um interpretador em C e comunicação com SGBDs.

- Versão 2 aparece em novembro de 1997, quando recebe seu nome inicial e é enriquecido com um interpretador de formulários (FI) - PHP/FI. Teve seu código fonte disponibilizado para a comunidade. Contava com 50.000 domínios que o utilizavam (em torno de 1% dos existentes na época).

- Versão 3 sai em seguida, logo em julho de 1998. Similar ao PHP atual, esta versão foi totalmente reescrita por Andi Gutmans e Zeev Suraski, programadores israelenses. Inicia o suporte à orientação a objetos e a sua extensibilidade, que atraiu muitos programadores. Rebatizado de PHP: Hypertext Processor. Já estava presente em 10% dos servidores web da Internet.

- Versão 4 sai em maio de 2000. Melhora de performance, suporte a muitos servidores web, a session, entre outros. Já está presente em 20% dos domínios da Internet.

- Versão 5 sai em julho de 2004. Seu foco principal é a orientação a objetos, que corrije deficiências e traz novos e amplos recursos para a orientação a objetos.

Delimitadores do PHP

Para que o interpretador reconheça quando tem que interpretar o script em PHP e quando não, usa-

se delimitadores para que quando os encontre ele interprete corretamente. Quando ele encontra o delimitador <?php ele começa a processar como PHP e quando encontra ?> ele pára de processar e tudo o mais envia para o browser como está. Existem outros tipos de delimitadores do PHP mas estes <?php ... ?> são os recomendados, por serem os mais abrangentes e terem suporte inclusive a XML.

Grandes Forças do PHP

- Sua simplicitade é muito importante para o programador que se inicia no PHP.

- Seus recursos atendem ao programador experiente.

- Sua documentação rica em conteúdo e exemplos facilita a solução de problemas, além da busca online.

- O suporte de maneira simples à maioria dos SGBDs do mercado atende às necessidades de sites dinâmicos com acesso a bancos de dados.

O Que é Possível Realizar em PHP?

- Em sendo uma linguagem de uso geral, podemos realizar praticamente qualquer atividade realizada com outra linguagem. Ele roda no lado servidor, como aplicação Web e também roda no lado cliente, com o PHP-GTK.

- Existem edições para os principais sistemas operacionais (Linux, BSDs, Windows, Mac OS X, etc).

- Suportado pela maioria dos servidores Web, especialmente pelo Apache.

- Bom suporte à orientação a objetos, em especial com a chegada da versão 5.

- Comunicação com Java.

- Suporte aos principais SGBDs do mercado (MySQL, PostgreSQL, Oracle, SQL Server, etc), atualmente são mais de vinte os suportados.

- Suporte a LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows somente), a IRC, compressão (gzip, bz2, rar), PDF, GD, Flash, XML, etc..

- TUTORIAL SIMPLES

Para continuar você deve ter um servidor web instalado e com suporte a PHP e, caso queira usar bancos de dados, instale um SGBD.

Os arquivos em PHP somente funcionam corretamente no navegador, se estivem no diretório reconhecido pelo servidor web, que é o DocumentRoot. Considerando que o PHP e Apache tenham sido instalados com o Xampp e que o diretório web seja o default, este fica em:

C:\Arquivos de Programas\Xampp\htdocs

Além disso qualquer script que contenha algo em PHP deve ter a extensão .php para que seja

corretamente processado pelo Apache.

Criar um arquivo chamado olamundo.php na pasta C:\Arquivos de Programas\Xampp\htdocs, com o conteúdo:

<?php echo "Olá PHP!"; ?>

Startar o Apache e abrir no browser assim:

http://127.0.0.1/olamundo.php

Visualizar resultado de script PHP no Browser

Após ter executado o olamundo.php no Browser solicite a visualização do código fonte da página exibida.

Veja que nada chega de PHP. O PHP é processado no servidor e envia apenas o resultado em HTML para o browser.

Alerta sobre Editores e Processadores de Texto com PHP Para criar scripts PHP evite o uso de processadores de texto e de editores como o Bloco de Notas, pois deixam sujeira no código e dificultam o salvamento do arquivo.

Preferentemente use uma IDE especializada em PHP, que inclusive tem outros recursos úteis, a exemplo do PHPEclipse ou um editor de texto como o freeware Win32pad.

Informações sobre o PHP

Para obter diversas informações úteis sobre o PHP instalado no seu servidor, execute um script com a função:

```
<?php
    phpinfo();
?>
```

Checar navegador em uso

```
if (strstr($_SERVER["HTTP_USER_AGENT"], "MSIE")) {
     echo "Seu navegador é o Internet Explorer!";
} else {
     echo "Seu navegador não é o IE!";
}
```

strstr - procura substrings.

Teste com echo e print

```
<?php
// Teste de echo e print
//print "primeiro, ", "segundo"; // Gera erro de parser
echo "primeiro, ", "segundo, ", "terceiro"; //Funciona
// Ou seja, print pode trabalhar apenas com uma string de cada vez e echo com
```

```
várias
?>
```

print retorna valor enquanto que echo não retorna.

```
<?php
$print = print "";
echo "Retorno do print: " . $print;
?>
```

Agora teste esse:

```
<?php
    $echo = echo "";
    echo "Retorno do echo: " . $echo;
?>
```

Irá receber um erro fatal (do parser).

- Configurações

php.ini

register_globals = off (por questões de segurança)

No Xampp o php.ini traz register_globals ativo por default.

GD, pgsql, mysql e outras extensões que queira usar deverá descomentar no php.ini, seção Extensions.

httpd.conf

Para poder alterar o diretório web default deverá mudar dois parâmetros no arquivo httpd.conf do Apache:

DocumentRoot "/opt/lampp/htdocs"

<Directory "/opt/lampp/htdocs">

Os diretórios acima são para a edição for Linux do Xampp. Para o Windows observe que o Apache utiliza, não barras, que são utilizadas neste SO, mas sim contra-barras e as vezes contra-barras duplas.

No caso queremos mudar o diretório default para /home/www, então alteramos o httpd.conf para que fique assim:

DocumentRoot "/home/www"

<Directory "/home/www">

No Windows fica assim:

DocumentRoot "c:/Arquivos de Programas/Xampp/htdocs"

<Directory "<c:/Arquivos de Programas/Xampp/htdocs">

Que devem ficar assim:

DocumentRoot "c:/1www"

<Directory "<c:/1www">

Configuração das extensões suportadas. Altere a linha:

DirectoryIndex index.html index.html.var index.php index.php3 index.php4

Veja que acima configurou-se para perceber inclusive arquivos com as extensões .php3 e .php4.

Caso não fossem incluídas, arquivos com extensão .php3 e .php4 não poderiam ser abertos neste servidor.

- Referência da Linguagem

Separador de Instruções

O ponto e vírgula ; é o separador de instruções em PHP (como em Perl e em C).

Sempre que uma instrução terminar devemos digitar um ponto e vírgula (echo "ola";).

Comentários

Em PHP podemos usar 3 tipos de comentários (/* ... */, // e #) mas devemos utilizar somente os dois primeiros e o primeiro é o mais eficiente, que é o /* ... */, que veio do C, já que # está em processo de obsolescência. Ou seja, devemos usar // ou /* ... */, de preferência este último.

- Tipos de Dados

O PHP suporta os oito tipos primitivos:

- boolean, integer, float e string (básicos) - array e object (compostos) - resource e NULL (especiais)

float é sinônimo de double em PHP.

Tipo de Variável

O tipo de uma variável em PHP não é controlado pelo programador, depende do valor da variável e é avaliado em tempo de execução. Não é permitido ao programador declarar o tipo de variáveis.

Funções que Retornal o Tipo

gettype

todas as is_type

Casting

(tipo) variavel;

Boleanos

Pode ser TRUE ou FALSE (case-insensitive)

Exemplos

```
<?php
echo gettype((bool) "")."<br>";        // bool(false)
echo gettype((bool) 1)."<br>";        // bool(true)
echo gettype((bool) -2)."<br>";        // bool(true)
echo gettype((bool) "foo")."<br>";    // bool(true)
echo gettype((bool) 2.3e5)."<br>";    // bool(true)
echo gettype((bool) array(12))."<br>"; // bool(true)
echo gettype((bool) array())."<br>";  // bool(false)
?>
```

Convertendo Explicitamente para Boleano

Usa-se o modificador (bool) ou (boolean).

Valores que são considerados FALSE

FALSE, 0, 0.0, "0", "", array vazio, objeto sem membros e NULL (e variáveis indefinidas).

Os demais são TRUE

Inteiros

Em PHP, inteiro é um número do conjunto matemático dos Inteiros (Z), que contem os negativos, o zero e os positivos.

Em PHP os inteiros podem ser decimais, octais ou hexadecimais.

octal - precedido por 0.

hexadecimal - precedido por 0x.

Exemplos

```
<?php
$a =1234;
echo $a."<br>"; // número decimal
$a =-123;
echo $a."<br>"; // um número negativo
$a =0123;
echo $a."<br>"; // número octal (equivalente a 83 em decimal)
$a =0x1A;
echo $a."<br>"; // número hexadecimal (equivalente a 26 em decimal)
?>
```

O tamanho dos inteiros depende da plataforma e é de 32 bits com sinal. O PHP não suporta inteiros sem sinal.

Overflow - caso seja especificado um número inteiro além dos limites, será interpretado como flutuante.

Convertendo Explicitamente para Inteiro

Usar o modificador (int) ou (integer).

Ou com a função intval().

De boleanos - FALSE será retornado como 0 e TRUE como 1.

De flutuantes - ao converter para inteiros serão truncados

De strings - A string será avaliada como um ponto flutuante se contiver qualquer um dos caracteres '.', 'e', ou 'E'. Em outros casos, ela será avaliada como um inteiro.

De outros tipos - não têm precisão, exatidão, portanto é melhor evitar.

Alerta

echo (int) ((0.1 + 0.7) * 10); // Exibirá 7 ao invés do esperado 8

Ponto Flutuante

É o float, double ou real.

Exemplos

1.234 ou 1.2e3 ou 7E-10

```
<?php
$a = 1.234;
echo $a."<br>";
$b = 1.2e3;
echo $b."<br>";
$c = 7E-4;
echo $c;
?>
```

O tamanho de um float depende também da plataforma e é de 64bits no formato IEEE(*).

Nunca compare números em ponto flutuante em igualdades, sob pena de cometer erros.

- - (Wikipedia - http://pt.wikipedia.org)O Instituto de Engenheiros Eletricistas e Eletrônicos ou IEEE (pronuncia-se I-3-E) é uma organização profissional sem fins lucrativos, fundada nos Estados Unidos. É a maior (em número de sócios) organização profissional do mundo. O IEEE foi formado em 1963 pela fusão do Instituto de Engenheiros de Rádio (IRA) com o Instituto Americano de Engenheiros Elétricistas (AIEE). O IEEE tem filiais em muitas partes do mundo, sendo seus sócios engenheiros eletricistas, engenheiros da computação, cientistas da computação, profissionais de telecomunicações etc. Sua meta é promover conhecimento no campo da engenharia elétrica, eletrônica e computação. Um de seus papéis mais importantes é o estabelecimento de padrões para formatos de computadores e dispositivos.

Strings

Em PHP um caractere ocupa um byte. Até a versão 5 o PHP não tem suporte a UNICODE, mas está previsto este suporte para a próxima versão (Fonte: Wikipedia - http://www.wikipedia.org).

Não há limite para o tamanho de uma string em PHP.

Especificando Strings

- apóstrofos (chamados de aspas simples ')

- aspas (chamadas de aspas duplas ")

- heredoc (<<<)

Exemplos

```
<?php
echo 'isto é uma string comum';
echo 'Você pode incluir novas linhas em strings,
dessa maneira que estará
tudo bem';

// Imprime: Arnold disse uma vez: "I\'ll be back"
```

```
echo 'Arnold once said: "I\'ll be back"';

// Imprime: Você tem certeza em apagar C:\*.*?
echo 'Você tem certeza em apagar C:\\*.*?';

// Imprime: Você tem certeza em apagar C:\*.*?
echo 'Você tem certeza em apagar C:\*.*?';

// Imprime: Isto não será substituido: \n uma nova linha
echo 'Isto não será substituido: \n uma nova linha';

// Imprime: Variaveis $também não $expandem
echo 'Variaveis $também não $expandem';

echo '<br>------------<br>';

$str = <<<EOD
Exemplo de uma string
distribuída em várias linhas
utilizando a sintaxe heredoc.
EOD;

/* Exemplo mais complexo, com variáveis */
class foo
{
   var $foo;
   var $bar;

   function foo()
   {
       $this->foo = 'Foo';
       $this->bar = array('Bar1', 'Bar2', 'Bar3');
   }
}

$foo = new foo();
$name = 'Meu nome';

echo <<<EOT
Meu nome é "$name". Eu estou imprimindo $foo->foo.
Agora, eu estou imprimindo {$foo->bar[1]}.
Isto deve imprimir um 'A' maiúsculo: \x41
EOT;

?>
```

Nulos em PHP

NULL

O valor especial NULL representa que a variável não tem valor. NULL é o único valor possível do tipo NULL.

A variável é considerada NULL se:

- ela foi assimilada com a constante NULL.

- ela ainda não recebeu nenhum valor ainda.
- ela foi apagada com unset().

Sintaxe

Há apenas um único valor do tipo NULL, e é a palavra (insensitiva ao caso) NULL.

Exemplos de uso:

Usando: empty() is_null() !isset()

```
$var = "";

empty($var) is true.
is_null($var) is false.
!isset($var) is false.
?>

<?php

//Outro exemplo

  $var = NULL;

  if (isnull("var")) {
    echo "var===NULL\n";
  } else {
    echo "var!==NULL\n";
  }

  if (isnull("test")) { // give FALSE, test is not set
    echo "test===NULL\n";
  } else {
    echo "test!==NULL\n";
  }

  $array['var'] = NULL;

  if (isnull("var", $array)) {
    echo "array['var']===NULL\n";
  } else {
    echo "array['var']!==NULL\n";
  }

  function isnull($var, $base = FALSE) {
    if ($base===FALSE) {
      $base = &$GLOBALS;
    } elseif (!is_array($base)) {
      return FALSE;
    }
    if ((array_key_exists($var, $base))&&($base[$var]===NULL)) {
      return TRUE;
    } else {
      return FALSE;
    }
  }
?>

:: NULL == NULL is true
```

```
:: NULL == FALSE is true
:: NULL == TRUE is false
```

Agora com SQL:

- NULL AND TRUE yields NULL
- NULL OR TRUE yields TRUE
- NULL AND FALSE yields FALSE
- NULL OR FALSE yields NULL
- NULL == TRUE yields FALSE
- NULL == FALSE yields FALSE

is_null

is_null - Informa se a variável é NULL

Descrição

bool is_null (mixed $var)

Retorna TRUE se var é null, FALSE senão.

Veja o tipo NULL para saber quando a variável é considerada NULL e quando não.

Veja mais em NULL, is_bool(), is_numeric(), is_float(), is_int(), is_string(), is_object(), is_array(), is_integer() e is_real().

unset

unset - Destrói a variável especificada

Descrição

void unset (mixed $var [, mixed $var [, $...]])

unset() destrói a variável especificada. Lembrando que no PHP 3, unset() sempre retornará TRUE (atualmente, o valor inteiro 1). No PHP 4, porém, unset() não é uma função verdadeira: agora é um construtor da linguagem. Como tal, nenhum valor é retornado, se tentar pegar o valor de unset() um erro de sintaxe (parse error) irá acontecer.

Exemplo:

```
<?php
// destrói uma única variável
unset ($foo);

// destrói um único elemento de uma matriz
unset ($bar['quux']);

// destrói mais de uma variável
unset ($foo1, $foo2, $foo3);
?>
```

- Variáveis

Em PHP as variáveis são iniciadas por um sinal de dólar $.

Variáveis Predefinidas

São as que já vêm definidas no próprio PHP. A função phpinfo() mostra também estas variáveis.

Com o PHP 4.2 o valor default da diretiva register_globals passou a ser off. Com isso as variáveis passaram a ser acessadas de forma diferente e muitos scripts deixaram de funcionar quanto tiveram o PHP atualizado e outros novos scripts não funcionaram devido esperar uma semelhante a anteerior.

```
On                                                                  Off

$DOCUMENT_ROOT                                    $_SERVER['DOCUMENT_ROOT'];
$HOME $_ENV['HOME'];
$GLOBALS
$_SERVER
$_GET
$_POST
$_REQUEST
$_SESSION
```

Obs.: Agora, como o default do PHP é register_globals = Off, faz-se necessário usar $_POST['nomecampo'], para receber o valor de um campo de form em script PHP.

Escopo de Variáveis

O escopo de uma variável é o contexto onde ela foi definida e geralmente o escopo é local.

```
$a = 1;
include ("teste.php");

// $a estará disponível, será vista por teste.php, pois foi definida antes.

include ("teste.php");
$a = 1;

// Aqui, como $a foi definida após a inclusão, não será visto pelo teste.php
```

Escopo de variáveis em funções

A palavra-chave global pode preceder uma variável para tornar seu escopo global, como também $GLOBALS[].

Exemplos:

```
<?php
$a = 1; /* escopo global */

function Teste(){
   echo $a; /* referencia uma variável do escopo local (não definida) */
}

Teste();
?>

<?php
$a = 1; /* escopo global */

function Teste(){
    global $a;
    echo $a; /* referencia a variável do escopo global */
}

Teste();
?>
```

Usando $GLOBALS no lugar de global

```
<?php
$a = 1;
$b = 2;

function Soma(){
   $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"];
}

Soma();
echo $b;
?>
```

Utilizando Variáveis Estáticas

Variáveis estáticas guardam o valor de variáveis entre execuções de funções.

Ao fechar o programa, ao fechar ou atualizar o browser, o valor da variável se perde.

Exemplos

```
<?php
function Teste (){
   $a = 0;
   echo $a;
   $a++;
}
```

```
for ($x=1;$x<10;$x++){
    Teste();
}
echo "<br><br>";

function Teste2(){
   static $a = 0;
   echo $a;
   $a++;
}

for ($x=1;$x<10;$x++){
    Teste2();
}

echo "<br><br>";

// Função recursiva
function Teste3()
{
   static $count = 0;

   $count++;
   echo $count;
   if ($count < 10) {
       Teste3 ();
   }
   $count--;
}

for ($x=1;$x<5;$x++){
    Teste3();
    if ($x < 4) echo " - ";

}

echo "<br><br>";
//Declarando variáveis static

function foo(){
   static $int = 0;          // correro
   //static $int = 1+2;        // errado (é uma expressão)
   //static $int = sqrt(121);  // wrong  (é uma expressão também)

   $int++;
   echo $int;
}

foo();

?>
```

Variáveis Variáveis

São variáveis cujos nomes podem ser criados dinamicamente.

Variável comun -> $variavel;

Variável variável -> $$variavelvariavel;

Ela torna o valor de uma variável e o trata como se fosse o nome de uma variável.

Obs.: variáveis variáveis não podem ser utilizadas com os arrays superglobais.

Determiando o Tipo das Variáveis

gettype

is_array, is_float, is_int, is_object, is_string, is_numeric

- Constantes

O valor de uma constante não pode ser alterado durante a execução do script.

Convenciona-se usar o nome de constantes com todas as letras em maiúsculas.

define ("NOME", "valor");

Exemplos

```
<?php
// Nomes de constantes válidos
define("FOO",    "alguma coisa");
define("FOO2",    "alguma outra coisa");
define("FOO_BAR", "mais alguma outra coisa")

// Nomes de constantes inválidas
define("2FOO",    "alguma coisa");

// Isto é válido, mas deve ser evitado:
// O PHP pode vir a fornercer uma constante mágica
// que danificará seu script
define("__FOO__", "alguma coisa");

?>
```

Obs.: Somente dados escalares (bolean, integer, float, e string) podem ser armazanados nos valores de constantes.

A função constant() retorna o valor de uma constante.

A função get_defined_constantes() retorna todas as constantes definidas.

Enquanto que defined() checa se uma constante foi definida.

```
define (NOME, valor);
define ("PESO", 70);
print "O peso vale " . PESO . " KG";

<?php
switch (PHP_OS){
    case "WIN32":
        echo "Sistema Windows";
        break;
    case "Linux":
        echo "Sistema Linux";
        break;
    case "OS/2":
        echo "Sistema OS/2";
        break;
    default:
        echo "Sistema não é Windows, Linux nem OS/2";
        break;
}
?>

<?php
if (strtoupper(substr(PHP_OS, 0, 3)) === 'WIN') {
    echo 'Este é um servidor usando Windows!';
} else {
    echo 'Este é um servidor que não usa Windows!';
}
?>
```

Constantes Mágicas

Essas são as cinco constantes "mágicas", que mudam dependendo de onde elas são utilizadas. Por exemplo, o valor de __LINE__ depende do número da linha em que é utilizada em seu script. Essas constantes especiais são insensíveis ao caso:

Algumas constantes "mágicas" do PHP

Nome Descrição

__LINE__ A linha atual do script. __FILE__ O caminho completo e nome do arquivo. Se utilizado dentro de um include, o nome do arquivo incluído será retornado. __FUNCTION__ O nome da função (Acrescentado no PHP 4.3.0). A partir do PHP 5 esta constante retorna o nome da função como ela foi declarada (sensível a maiúsculas e minúsculas). No PHP 4 sempre retorna o nome em minúsculas. __CLASS__ O nome da classe (Acrescentado no PHP 4.3.0). A partir do PHP 5 esta constante retorna o nome da função como ela foi declarada (sensível a maiúsculas e minúsculas). No PHP 4 sempre retorna o nome em minúsculas. __METHOD__ O nome do método de classe. (Acrescentado no PHP 5.0.0). O nome do método é retornado como foi declarado (sensível a maiúsculas e minúsculas).

Veja também get_class(), get_object_vars(), file_exists(), e function_exists().

- Expressões

Uma expressão é qualquer coisa que tem um valor, normalmente na forma de constantes ou variáveis.

$a = 5;

Temos acima uma equação formada por duas expressões, a da esquerda composta por uma variável e a da direita composta por uma constante.

Funções são expressões cujo valor é igual ao seu valor de retorno.

O PHP é uma linguagem orientada as expressões.

Atribuições são analisadas da direita para a esquerda.

$b = $a = 5; // $a recebe 5 e então $b recebe 5

Exemplo

```
<?php
function dobro($i){
   return $i*2;
}
$b = $a = 5;echo $b."<br>";        /* atribui o valor cinco às variáveis $a e $b
*/
$c = $a++;echo $c."<br>";          /* pós-incremento, atribui o valor original
de $a
                        (5) para $c */
$e = $d = ++$b;echo $e."<br>";    /* pré-incremento, atribui o valor
incrementado de
                        $b (6) a $d e $e */

/* neste ponto, tanto $d quanto $e são iguais a 6 */

$f = dobro($d++);echo $f."<br>";  /* atribui o dobro do valor de $d antes
                        do incremento, 2*6 = 12 a $f */
$g = dobro(++$e);echo $g."<br>";  /* atribui o dobro do valor de $e depois
                        do incremento, 2*7 = 14 a $g */
$h = $g += 10;echo $h."<br>";      /* primeiro, $g é incrementado de 10 e
termina com o
                        valor 24. o valor da atribuição (24) é
                        então atribuído a $h, e $h termina com o valor
                        24 também. */
?>
```

- Operadores

Um operador é algo que alimentado com um ou mais valores devolve outro valor. Assim as funções e outros construtores que retornam valor são operadores.

Tipos de Operadores

- unários - operam apenas em um valor (!, ++, ...)

- binários - operam em dois valores (+, - , ...)

- ternário - selecionar entre 2 valores, dependendo de um terceiro. Emglobá-los entre parênteses facilita o entendimento.

Precedência de Operadores

A precedência de um operador em relação a outro diz que um operador deve ser executado antes do outro.

Exemplo

1 + 5 * 3

Dependendo da precedência dos operadores + ou * teremos resultados diferentes. Se o operador + tiver precedência superior ao * então resolveremos assim:

Somamos 1 + 5 e somente depois multiplicamos por 3 -> (1+5)*3 = 6*3 = 18

Este resultado está incorreto, pois o operador * tem precedência superior ao + e deveria ser assim:

1+(5*3) = 1+15=16

Para facilitar a percepção das precedências usa-se parêntesis

1 + (5 * 3) = Sempre devemos resolver antes o que estiver entre parêntesis.

1 + 15 = 16 (Neste caso fica mais claro).

Veja a tabela contendo a precedência de dos operadores do PHP, da maior precedência para a menor:

- e / têm precedência sobre + e -.

Obs.: Para uma relação completa das precedências veja o manual oficial no site do PHP.

Operadores Aritméticos

São os operadores correspondentes às quatro operações comuns da matemática adicionados ao operador módulo:

• soma (+) • subtração (-) • multiplicação (*) • divisão (/) • módulo (%)

O módulo é o resto de um número dividido por outro.

Exemplos de operadores aritméticos

```
$a = 9;
$b = 4;
echo "\$a + \$b = ".($a + $b); // 13
echo "\$a + \$b = ".$a + $b; // Retorna 4, pois após o ponto é considerado
strings
echo "\$a - \$b = ".($a - $b); // 5
echo "\$a * \$b = ".$a * $b; // 36
echo "\$a / \$b = ".$a / $b; // 2.25 - Divisão Quociente de $a por $b.
echo "\$a % \$b = ".$a % $b; // 1 - Resto de $a por $b
```

Operadores de Atribuição

O operador básico de atribuição é o sinal de igualdade =.

Exemplo

```
$x = $y + 3; // O que representa: 3 será adicionado a $y e o resultado será
atribuído a $x
$a = 3;
$a += 5; // Que é semelhante a $a = $a + 5;
$b = "Bom ";
$b .= "Dia!"; // Similar a $b = $b . "Dia!";
```

Exemplos

```
<?php
$a = 3;
$a += 5; // $a recebe 5 e soma com seus 3, tipo: $a = $a + 5;
echo "\$a vale " . $a;

$b = "Bom ";
$b .= "Dia!"; // $b fica com "Bom Dia!", como em $b = $b . "Dia!";
echo "\$b vale " . $b;

$c=2;
$a -= $c;    //$a = $a - $c    Subtração
echo "\$a -= \$c vale " . $a;
$a *= $c;    //$a = $a * $c    Multiplicação
echo "\$a *= \$c vale " . $a;
$a /= $c;    //$a = $a / $c    Divisão
echo "\$a /= \$c vale " . $a;
$resto = $a % $c;    //$a = $a % $c    Módulo (resto)
echo "Resto de $a % $c vale: " . $resto;
?>
```

Operadores de Controle de Erro

Representado pelo símbolo de arroba @. Quando precede uma expressão ele abafa as mensagens de

erro.

Observação - Somente funciona em expressões, ou seja, em qualquer construção que retorne algo.

Recomendação - Deixar todas as mensagens de erro originais do PHP livres em ambientes de desenvolvimento (de testes). Somente utilizar @ em ambiente de produção, após executar os testes.

Exemplos

```
<?php
/* Erro intencional */
$a = 6;
$b = 0;

echo "Camuflando erro de divisão por zero";
$c = @($a / $b);
?>
```

Operadores de Execução

Existem algumas funções de execução de programas (shell_exec e outras) como também existe um operador, que é formado por dois sinais de crase` `.

Nota: Caso safe_mode esteja desativado no php.ini como também shell_exec() então o operador de execução também fica dasativado.

Exemplo

```
<?php
// Em PHP a crase ` é um operador de execução de arquivos do SO
// Como em scripts bash

// Exibir todos os arquivos do diretório atual, inclusive os ocultos
if (PHP_OS == "WINNT"){

    $output = `dir/o/p`;
    echo "<pre>$output

";
} elseif (PHP_OS == "Linux){
   $output = `ls -la`;

echo "
$output

";
}else{
   echo "Você está usando um SO diferente de Linux e de Windows!"

} ?>
```

Operadores de Incremento e de Decremento

Os operadores de pré e pós-incremento/decremento são suportados pelo PHP.

```
++$a (pré-incremento) - Primeiro incrementa $a de 1, depois retorna $a
incrementado
$a++ (pós-incremento) - Primeiro retorna $a, depois incrementa $a de 1
--$a (pré-decremento) - Primeiro decrementa $a de 1, depois retorna $a
decrementado
$a-- (pós-decremento) - Primeiro retorna $a, depois decrementa $a de 1
```

Exemplos:

```
<?php
echo "<h3>Pós-incremento</h3>";
$a = 5;
echo "\$a = ".$a."<br><br>";
echo "\$a++ deve ser: " . $a++ . "<br />\n";
echo "\$a deve ser: " . $a . "<br />\n";

echo "<h3>Pré-incremento</h3>";
$a = 5;
echo "++\$a deve ser: " . ++$a . "<br />\n";
echo "\$a deve ser: " . $a . "<br />\n";

echo "<h3>Pós-decremento</h3>";
$a = 5;
echo "\$a-- deve ser: " . $a-- . "<br />\n";
echo "\$a deve ser: " . $a . "<br />\n";

echo "<h3>Pré-decremento</h3>";
$a = 5;
echo "--\$a deve ser: " . --$a . "<br />\n";
echo "\$a deve ser: " . $a . "<br />\n";
?>
```

Operadores Lógicos

Utilizados para comparar duas expressões e o resultado será TRUE ou FALSE.

Exemplos:

```
<?php
$a = true; $b = FALSE; // true e false são insensitivos

echo ($a and $b)? "T<br>":"F<br>";    //E    Verdadeiro (TRUE) se tanto $a
quanto $b são verdadeiros.
echo ($a or $b)? "T<br>":"F<br>";    //OU    Verdadeiro se $a ou $b são
verdadeiros.
echo ($a xor $b)? "T<br>":"F<br>";    //XOR    Verdadeiro se $a ou $b são
verdadeiros, mas não ambos.
```

```
echo (! $a)? "T<br>":"F<br>";        //NÃO    Verdadeiro se $a não é verdadeiro.
echo ($a && $b)? "T<br>":"F<br>";    //E    Verdadeiro se tanto $a quanto $b são
verdadeiros.
echo ($a || $b)? "T<br>":"F<br>";    //OU    Verdadeiro se $a ou $b são
verdadeiros.
?>
```

Operadores de String

Strings em PHP são concatenadas com o operador ponto final ".".

Exemplos:

```
<?php
$a = "Olá ";
$b = $a . "mundo do PHP!";
echo $b;

$a = "Olá ";
$a .= "meu mundo!";

echo "<br>" . $a;
?>
```

Convertendo strings em números

```
<?php
$foo = 1 + "10.5";echo $foo."<br>";             // $foo é float (11.5)
$foo = 1 + "-1.3e3";echo $foo."<br>";              // $foo é float (-1299)
$foo = 1 + "bob-1.3e3";echo $foo."<br>";          // $foo é integer (1)
$foo = 1 + "bob3";echo $foo."<br>";                // $foo é integer (1)
$foo = 1 + "10 Small Pigs";echo $foo."<br>";      // $foo é integer (11)
$foo = 4 + "10.2 Little Piggies";echo $foo."<br>"; // $foo é float (14.2)
$foo = "10.0 pigs " + 1;echo $foo."<br>";          // $foo é float (11)
$foo = "10.0 pigs " + 1.0;echo $foo."<br>";        // $foo é float (11)
?>
```

Operações com Strings

```
<?php
// Pega o primeiro caracter da string
$str = 'Isto é um teste.';
$first = $str{0};
echo $first."<br>";
// Pega o terceiro caracter da string
$third = $str{2};
echo $third."<br>";
// Pega o último caracter da string
$str = 'Isto ainda é um teste.';
```

```
$last = $str{strlen($str)-1};
echo $last."<br>";
// Modifica o ultimo caracter da string
$str = 'Olhe o mal';
echo $str{strlen($str)-1} = 'r';
?>
```

Operadores com Arrays

Exemplo Nome Resultado

$a + $b União União de $a e $b.

$a == $b Igualdade TRUE se $a e $b tem os mesmos elementos.

$a === $b Identidade TRUE se $a e $b tem os mesmos elementos na mesma ordem.

$a != $b Desigualdade TRUE se $a não é igual a $b.

$a <> $b Desigualdade TRUE se $a não é igual a $b.

$a !== $b Não identidade TRUE se $a não é identico a $b.

O operador + acrescenta o array da direita no array da esquerda, contudo, chaves duplicadas NÃO são sobrescritas.

Exemplos:

```
<?php
$a = array("a" => "maçã", "b" => "banana");
$b = array("a" =>"pêra", "b" => "framboesa", "c" => "morango");

$c = $a + $b; // Uniao de $a e $b
echo "União de \$a e \$b: \n";
var_dump($c);

$c = $b + $a; // União de $b e $a
echo "União de \$b e \$a: \n";
var_dump($c);
?>
```

Observar que na união de $a+$b o valor de "b" é banana em $a, ele não foi sobrescrito por framboesa de $b.

Assim como framboesa "b" em $b não foi substituído por banana de $a na união de $b+$a.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/PHP/PHP1"
Page categories: PHP

Aplicativos em PHP/Referências/PHP/PHP2

De Wikibooks

< Aplicativos em PHP | Referências | PHP

Conteúdo

- 1 Continuação
- 2 ALERTA - PHP4 chegando ao fim da linha
- 3 - Estruturas de Controle
  - 3.1 if
  - 3.2 else
  - 3.3 elseif
  - 3.4 while
  - 3.5 do ... while
  - 3.6 for
  - 3.7 foreach
  - 3.8 break
  - 3.9 continue
  - 3.10 switch
- 4 Estruturas em Obsolescência
- 5 Formatando Números
- 6 Validação de Ano Bisexto
- 7 Algumas Funções Matemáticas
- 8 Enviar E-mail via PHP
- 9 Integração entre PHP e JavaScript
  - 9.1 PHP Comunicando com JavaScript: PHP recebendo variáveis do JavaScript e JavaScript recebendo variáveis do PHP
  - 9.2 Outro Exemplo de Integração do PHP com JavaScript
- 10 PHP com banco de dados Access
- 11 Sugestão de Livros
- 12 Links

ALERTA - PHP4 chegando ao fim da linha

Alerta para que quem ainda não migrou para o PHP 5 inicie sua migração:

Depois de 3 anos de PHP5 e com o PHP 6 chegando, a versão 4 será descontinuada a partir do ano que vem. Bugs críticos ainda serão corridos até o final de agosto de 2008.

Enviado por Marcos Alexandre Lemos Rodrigues para o br-linux (marcosalexandre·rodriguesΘgmail·com)

Notícia original em ingles:

PHP 4 end of life announcement [13-Jul-2007]

Today it is exactly three years ago since PHP 5 has been released. In those three years it has seen many improvements over PHP 4. PHP 5 is fast, stable & production-ready and as PHP 6 is on the way, PHP 4 will be discontinued.

The PHP development team hereby announces that support for PHP 4 will continue until the end of this year only. After 2007-12-31 there will be no more releases of PHP 4.4. We will continue to make critical security fixes available on a case-by-case basis until 2008-08-08. Please use the rest of this year to make your application suitable to run on PHP 5.

For documentation on migration for PHP 4 to PHP 5, we would like to point you to our migration guide. There is additional information available in the PHP 5.0 to PHP 5.1 and PHP 5.1 to PHP 5.2 migration guides as well.

Documentos para ajudar na migração:

Migration Guide - http://www.php.net/manual/en/migration5.php Também - http://www.php.net/manual/en/migration51.php e http://www.php.net/manual/en/migration52.php

- Estruturas de Controle

As principais estruturas para controlar o fluxo dos scripts PHP.

if, else, elseif, while, do... while, for, break, consinue e switch.

Um script PHP é formado por instruções (cada instrução termina com ;).

Uma instrução pode ser:

```
- uma atribuição
- uma chamada de função
- um laço (loop)
- uma instrução condicional
- até uma instrução nula (;;).
```

As instruções podem ser agrupadas com chaves, formando blocos.

if

Uma construção muito importante e versátil. Permite a execução condicional de fragmentos de código.

Sintaxe:

```
if (expressao) {
    instrucoes;
}
```

expressao - é avaliada como TRUE ou FALSE

```
   - Se TRUE as instrucoes serão executadas

   - Se FALSE as instrucoes serão ignoradas
```

Exemplos:

```
<?php
$a = 5;
$b = 3;
if ($a > $b){
   echo "a é maior que b";
}
?>
```

Muito cuidado:

```
<?php
$a = 3;
$b = 5;
if ($a = $b){
   echo "a é maior que b";
}
?>
```

Sempre exibirá "a é maior que b", mesmo que não seja. Por que?

Porque o teste está errado. Está apenas atribuindo o valor de $b a $a. O teste correto deve ser assim:

if ($a == $b) //Checa apenas se ambas as variáveis têm o mesmo valor

ou então assim:

if ($a === $b) //Checa se ambas as variáveis têm o mesmo valor e também se são do mesmo tipo

Veja este exemplo:

if (1 == "1") retornará TRUE

mas

if (1 === "1") retornará FALSE, pois 1 é numérico e "1" é string.

else

Executa instrucoes2 caso expressao seja FALSE.

Sintaxe:

```
if (expressao) {
    instrucoes;
} else {
    instrucoes2;
}
```

Exemplos:

```
<?php
$a = 3;
$b = 5;
if ($a > $b){
   echo "a é maior que b";
} else {
   echo "a NÃO é maior que b";
}
?>
```

elseif

É uma combinação de if com else. Caso a expressao do if seja FALSE então o elseif testa a expresao2, se esta for TRUE instrucoes2 serão executadas, caso contrário instrucoes3 serão executadas.

Sintaxe:

```
if (expressao) {
    instrucoes;
} elseif (expressao2) {
    instrucoes2;
} else {
    instrucoes3;
}
```

```
<?php
$a = 3;
$b = 3;
if ($a > $b){
   echo "a é maior que b";
} elseif ($a==$b){
   echo "a é igual a b";
} else {
   echo "a NÃO é maior que b nem igual a b";
}
?>
```

while

Executa instruções várias vezez, enquanto uma expressão for verdadeira (TRUE).

Sintaxe:

```
while (expressao) {
    instrucoes;
}
```

Em cada iteração expressao é avaliada, se TRUE instrucoes serão executadas, se FALSE, serão ignoradas.

Exemplos:

```
<?php

$i = 1;
while ($i <= 10) {
   echo $i++;  /* o valor impresso será
                  $i depois do acréscimo
                  (post-increment) */
}

?>
```

do ... while

Semelhante ao while, diferindo no fato da expressão condicional ficar ao final das iterações.

No do ... while a primeira iteração sempre executará as instruções incondicionalmente.

Sintaxe:

```
do {
    instrucoes;
} while (expressoes);    // Atentar para o ponto e vírgula ao final
```

Exemplos:

```
<?php
// Executa pelo menos uma vez incondicionalmente
$i = 6; $fator = 2; $minimo = 10;
do {
   if ($i < 5) {
       echo "\$i não é grande o suficiente";
       break;
   }
   $i *= $fator;
   if ($i < $minimo) {
       break;
   }
   echo "\$i está Ok e vale " . $i;

} while (0);
```

```
/* Exemplo simples
$i = 0;
do {
   echo $i;
} while ($i > 0);
*/
?>
```

for

Laço composto por 3 instruções. A primeira é a de inicialização da variável, a segunda será avaliada a cada iteração e a terceira é a de incremento. As 3 separadas por ponto e vírgula. Todas as 3 são opcionais e caso a segunda seja nula o loop será infinito.

Sintaxe:

```
for (expr1; expr2; expr3) {
    instrucoes;
}
```

- expr1 será avaliada uma única vez incondicionalmente;

- expr2 será avaliada no início de cada iteração:

```
   - Se TRUE o laço continua e as instruções serão executadas

   - Se FALSE o laço termina.
```

- expr3 é avaliada ao final de cada iteração, no caso incrementa a variável

Exemplos:

```
<?php
/* exemplo 1 - Controla o fluso no for*/
echo "<br><br>1- ";
for ($i = 1; $i <= 10; $i++) {
   echo $i;
}
echo "<br><br>2- ";
/* exemplo 2 - Controle o fluxo no if interno*/
for ($i = 1; ; $i++) {
   if ($i > 10) {
       break;
   }
   echo $i;
}
echo "<br><br>3- ";
/* exemplo 3 - Controle o fluxo no if interno*/
$i = 1;
for (; ; ) {
   if ($i > 10) {
       break;
   }
   echo $i;
   $i++;
```

```
}
echo "<br><br>4- ";
/* exemplo 4 */
for ($i = 1; $i <= 10;$i++);
    echo $i;
?>
```

foreach

Laço para varrer os elementos de uma matriz (array).

Atentar para o fato de que o foreach não requer reset() antes, pois quando é executado o ponteiro interno do array já é zerado automaticamente.

Sintaxe:

```
foreach (expressao_array as $valor){
 instrucoes;
}

foreach (expressao_array as $chave => $valor) {
 instrucoes;
}
```

Exemplos:

```
<?php
//Você pode ter notado que os seguintes itens são funcionalmente idênticos:
$arr = array("um", "dois", "três");
reset ($arr); // Aponta para o primeiro elemento
while (list(, $value) = each ($arr)) {
   echo "Valor: $value<br />";
}
foreach ($arr as $value) {
   echo "Valor: $value<br />";
}

//Os seguintes também são funcionalmente idênticos:
$arr = array("one", "two", "three");
reset($arr);
while (list($key, $value) = each ($arr)) {
   echo "Chave: $key; Valor: $value<br />";
}
foreach ($arr as $key => $value) {
   echo "Chave: $key; Valor: $value<br />";
}

/* exemplo foreach 1: somente valores */
$a = array(1, 2, 3, 17);
foreach ($a as $v) {
   echo "Valor atual de \$a: $v.<br>";
}

/* exemplo foreach 2: valores (com as chaves impressas para ilustração) */
$a = array(1, 2, 3, 17);
$i = 0; /* para exemplo somente */
```

```
foreach ($a as $v) {
   echo "\$a[$i] => $v.<br>";
   $i++;
}

/* exemplo foreach 3: chaves e valores */
$a = array (
   "um" => 1,
   "dois" => 2,
   "três" => 3,
   "dezessete" => 17
);
foreach ($a as $k => $v) {
   echo "\$a[$k] => $v.<br>";
}
?>
```

break

Cancela a execução de laços: for, foreach, while, do ... while ou switch.

break n é suportado, onde n é o número de estruturas a serem canceladas.

Sintaxe:

```
break;

break n;
```

Exemplos:

Exemplo simples

```
for ($x = 1; $x < 10; $x++){
    if($x == 5){
        break;
    }
    echo $x;
}
```

Imprimirá 1234, pois quando $x for 5 sairá do laço.

Exemplo sofisticado

```
<?php
$arr = array('um', 'dois', 'três', 'quatro', 'PARE', 'cinco');
while (list (, $val) = each ($arr)) {
   if ($val == 'PARE') {
       break;    /* Você poderia colocar 'break 1;' aqui. */
   }
   echo "$val<br />";
}
echo "<br />";
/* Utilizando o argumento opcional. */

$i = 0;
```

```
while (++$i) {
   switch ($i) {
   case 5:
       echo "No 5<br />";
       break 1;  /* Sai somente do switch. */
   case 10:
       echo "No 10; saindo<br />";
       break 2;  /* Sai do switch e while. */
   default:
       break;
   }
}
?>
```

continue

Sai da atual iteração de um loop para continuar na próxima iteração.

Sintaxe:

```
continue;

continue n;
```

Enquanto o break encerra definitivamente um laço, o continue encerra somente a iteração atual.

Exemplos:

Exemplo simples

```
for ($x = 1; $x < 10; $x++){
    if($x == 5){
        continue;
    }
    echo $x;
}
```

Imprimirá 12346789, pois quando $x for 5 sairá da iteração e continuará na próxima que é o 6.

Exemplo sofisticado

```
<?php
$arr = array(1,2,3,4,5,6,7,8);
while (list ($key, $value) = each ($arr)) {
   if (!($key % 2)) { // pula itens pares, ou seja, processa somente ímpares
       continue;
   }
   echo ($value);
}
echo "<br>";
$i = 0;
while ($i++ < 5) {
   echo "Fora<br />";
   while (1) {
       echo "  Meio<br />";
       while (1) {
           echo "  Dentro<br />";
           continue 3;
```

```
        }
        echo "Isto nunca será exibido.<br />";
    }
    echo "Nem isso.<br />";
}

// Outro exemplo
  for ($i = 0; $i < 5; ++$i) {
     if ($i == 2)
         continue
     print "$i\n";
  }
?>
```

switch

Similar a uma série de construções if seguidas.

Sintaxe:

```
switch (variavel) {
    case valor1:
        instrucoes;
        break;
    case valor2:
        instrucoes2;
        break;
    case valorN;
        instrucoesN;
        break;
    default:
        instrucoesDefault;
        break;
}
```

Exemplos:

```
<?php

$i = 1;

// Estrutura com if
if ($i == 0) {
   echo "\$i igual a 0";
} elseif ($i == 1) {
   echo "\$i igual a 1";
} elseif ($i == 2) {
   echo "\$i igual a 2";
}
echo "<br>";
// Estruturas com switch
switch ($i) {
   case 0:
       echo "\$i igual a 0";
       break;
   case 1:
       echo "\$i igual a 1";
       break;
```

```
   case 2:
       echo "\$i igual a 2";
       break;
}
echo "<br>";
$i = 2;
// Executará todos, falta o break
switch ($i) {
   case 0:
       echo "\$i igual a 0";
   case 1:
       echo "\$i igual a 1";
   case 2:
       echo "\$i igual a 2";
}
echo "<br>";
// Simulando intervalos
switch ($i) {
   case 0:
   case 1:
   case 2:
       echo "\$i é menor que 3 mas não negativo";
       break;
   case 3:
       echo "\$i é 3";
}
echo "<br>";
// Valor default
switch ($i) {
   case 0:
       echo "\$i igual a 0";
       break;
   case 1:
       echo "\$i igual a 1";
       break;
   case 2:
       echo "\$i igual a 2";
       break;
   default:
       echo "\$i não é igual a 0, 1 ou 2";
}
?>
```

Estruturas em Obsolescência

Devemos evitar, pois em futuras versões não mais serão utilizados.

Funções com nomes antigos, tipo:

pg_numrown, pg_fecharray e similares.

Caso se faça uma busca por estas funções no site do PHP nada será retornado.

As funções atuais tem nomes com espaço separando palavras, como:

pg_num_rown, pg_fech_array e similares.

Algumas funções agora tem outro nome, por exemplo:

pg_exec agora deve ser substuida por pg_query

Algumas estruturas de controle usavam os dois pontos:

while: ... endwhile, if: ... endif, devemos preferir:

while(){ ... }, if(){ ... }

Evitar o uso do comentário tipo shell #.

Estruturas do tipo $HTTP_POST_VARS e similares foram substituídas por: $_POST e similares.

Formatando Números

number_format

number_format -- Formata um número com os milhares agrupados

string number_format (float number [, int decimals])

string number_format (float number, int decimals, string dec_point, string thousands_sep)

number_format() retorna uma versão formatada de number. Esta função aceita um, dois ou quatro parâmetros (não três):

Se apenas um parâmetro é dado, number será formatado sem decimais, mas com uma virgula (",") entre cada grupo de milhar.

Se dois parâmetros são dados, number será formatado com o número de casas decimais especificadas em decimals com um ponto (".") na frente, e uma vírgula (",") entre cada grupo de milhar.

Se todos os quatro parâmetros forem dados, number será formatado com o número de casas decimais em decimals, dec_point ao invés do ponto (".") antes das casas decimais e thousands_sep ao invés de uma vírgula (",") entre os grupos de milhares.

Somente o primeiro caractere de thousands_sep é usado. Por exemplo, se você usar foo como o parâmetro thousands_sep no número 1000, number_format() irá retornar 1f000.

Exemplo 1. Exemplo number_format()

Por exemplo, a notação Francesa usa duas casas decimais, vírgula (',') como separador decimal, e espaço (' ') como separador de milhar. Isto é feito com a linha :

```
<?php

// string number_format ( float number, int decimals, string dec_point, string
thousands_sep )
$number = 1234.56;

'''// Notação Brasileira'''
$numero_format_brasil = number_format($numero, 2, ',', '.');

// 1.234,56
echo "O número' $number' no formato brasileiro fica
'$numero_format_brasil'<br><br>";
?>
```

Validação de Ano Bisexto

```
<?php
function ano_bisexto($ano){
    return ((($ano%4)==0 && ($ano%100) != 0) || ($ano%400)==0);
}

if(ano_bisexto(2006))
    echo "Ano bisexto";
else
    echo "Ano não bisexto";
?>
```

Algumas Funções Matemáticas

```
abs -- Valor absoluto
mixed abs ( mixed número )
<?php
$abs = abs(-4.2); // $abs = 4.2; (double/float)
$abs2 = abs(5);  // $abs2 = 5; (inteiro)
$abs3 = abs(-5);  // $abs3 = 5; (inteiro)
?>

ceil -- Arredonda frações para cima
float ceil ( float valor )

<?php
echo ceil(4.3);    // 5
echo ceil(9.999);  // 10
?>

floor -- Arredonda frações para baixo
float floor ( float valor )
<?php
echo floor(4.3);  // 4
echo floor(9.999); // 9
?>

max -- Localiza o maior número
mixed max ( number arg1, number arg2 [, number ...] )
mixed max ( array numbers [, array ...] )
<?php
echo max(1, 3, 5, 6, 7);  // 7
echo max(array(2, 4, 5)); // 5

echo max(0, 'hello');    // 0
echo max('hello', 0);    // hello
echo max(-1, 'hello');    // hello

// Com arrays múltiplos, max compara da esquerda para direita,
// assim nesse exemplo: 2 == 2, mas 4 < 5
$val = max(array(2, 4, 8), array(2, 5, 7)); // array(2, 5, 7)

// Se forem informados um array e um não array, o array
// é sempre retornado como se ele fosse o maior
```

```
$val = max('string', array(2, 5, 7), 42);  // array(2, 5, 7)
?>

min -- Localiza o menor número
mixed min ( number arg1, number arg2 [, number ...] )
mixed min ( array numbers [, array ...] )

<?php
echo min(2, 3, 1, 6, 7);  // 1
echo min(array(2, 4, 5)); // 2

echo min(0, 'hello');    // 0
echo min('hello', 0);    // hello
echo min('hello', -1);    // -1

// Com arrays múltiplos, min compara da esquerda para direita,
// assim nesse exemplo: 2 == 2, mas 4 < 5
$val = min(array(2, 4, 8), array(2, 5, 1)); // array(2, 4, 8)

// Se ambos forem um array e um não array, o array
// nunca será retornado porque ele é sempre considerado o maior
$val = min('string', array(2, 5, 7), 42);  // string
?>

count -- Conta o número de elementos de uma variável
int count ( mixed var [, int mode] )
<?php
$a[0] = 1;
$a[1] = 3;
$a[2] = 5;
$result = count($a);
// $result == 3

$b[0] = 7;
$b[5] = 9;
$b[10] = 11;
$result = count($b);
// $result == 3;
?>
```

Exemplo 2. Uso recursivo da função count() (PHP >= 4.2.0)

```
<?php
$food = array( 'fruits'  => array('orange', 'banana', 'apple'),
'veggie'  => array('carrot', 'collard','pea'));
// recursive count
echo count($food,COUNT_RECURSIVE);  // mostra 8
// normal count
echo count($food);                  // mostra2 2
?>

pow -- Potência
number pow ( number base, number exp )

var_dump( pow(2,8) ); // int(256)
echo pow(-1,20); // 1
echo pow(0, 0); // 1
echo pow(-1, 5.5); // erro
```

```
rand -- Gera um número aleatório
int rand ( [int min, int max] )

echo "Exibir 20 números aleatórios entre 20 e 160<br><br>";
for($x=20;$x<40;$x++){
    echo rand($x, 4*$x)."<br>";
}
exit("Saindo...");

round -- Arredonda um número
float round ( float val [, int precision] )

<?php
echo round(3.4);        // 3
echo round(3.5);        // 4
echo round(3.6);        // 4
echo round(3.6, 0);      // 4
echo round(1.95583, 2);  // 1.96
echo round(1241757, -3); // 1242000
echo round(5.045, 2);    // 5.04
echo round(5.055, 2);    // 5.06
?>

sqrt -- Raiz quadrada
float sqrt ( float arg )

<?php
// Precisão depende de sua diretiva precision
echo sqrt(9); // 3
echo sqrt(10); // 3.16227766 ...
?>
```

Forçando a Limpeza do Cache

```
<?php
// Este é para os servidores de proxy. Diz para baixar, bypassando o proxy
header ("Cache-Control: no-cache, must-revalidate");

// Este é para o navegador e nem sempre funciona (falta de padrão entre eles)
header ("Pragma: no-cache");
?>
```

Redirecionamento de Páginas

```
header("location: novapagina.php");
```

Exibir o conteúdo de um arquivo

readfile -- Lê e exibe o conteúdo de um arquivo

readfile('http://www.google.com.br');

readfile('/home/1www/pagina2.php');

Enviar E-mail via PHP

```
<?php
$to = "ribamar.sousa@dnocs.gov.br";
$subject="Apenas testando";
$message = "Estou testando o envio de e-mail pelo PHP.";
$email="ribafs@gmail.com";
$ret=mail($to, $subject, $message, "From: $email\r\nReply-to: $email\r\n");
echo $ret;
?>
```

Usando Favicon em sites

Aquele pequeno ícone que fica à esquerda da URL do site, na caixa Location do Browser, que inclusive personaliza a barra de links quando arrastamos o endereço do site. Para que nosso site apareça para o visitante com um link, devemos seguir os seguintes procedimentos:

- Criar uma imagem com o logo do site, no formato png, jpg, gif, etc com 32x32 pixels

- Salvar como favicon.png ou outra extensão

- Adicionar a TAG abaixo, na TAG head, como abaixo:

```
<head>
...
<link href="http://www.seusite.com.br/diretorio/favicon.png" type="image/gif"
rel="icon">
...
</head>
```

Também pode ser assim:

rel="shortcut icon"

Este ícone pode ser inclusive animado.

Referência: lista da Dicas-L (http://www.dicas-l.com.br).

Integração entre PHP e JavaScript

O PHP pode ter acesso às variáveis do JavaScript e o JavaScript pode ter acesso às variáveis do PHP. Veja alguns exemplos:

PHP Comunicando com JavaScript: PHP recebendo variáveis do JavaScript e JavaScript recebendo variáveis do PHP

```
<!-- Arquivo php2_js.php -->

<h1 style='font-size:18px;text-align:center;'>Integração entre PHP e
JavaScript</h1>

Lembrando que o PHP é processado no servidor (pelo Apache, por exemplo)

Enquanto que o JavaScript é processado no cliente (pelo Navegador).
```

```
<h2>Variáveis em JavaScript</h2>

<script>
var variavel_js=250;
alert("A variável_js vale "+variavel_js);
document.write("A variável_js vale "+variavel_js);
document.write("<br>");
document.write("Aqui não saimos do Navegador ainda<br>");
</script>

<h2>Variáveis em PHP</h2>

<?php
$variavel_php=850;
print "<script>alert('A variavel_php vale '+$variavel_php)</script>";
print "O alert anterior recebeu a variável do servidor e exibiu<br>";
print "<br>A \$variavel_php vale $variavel_php<br>";
?>

<h2>JavaScript recebendo de PHP</h2>

<script>
variavel2_js = '<?=$variavel_php?>' - 150;

alert("variavel2_js recebeu \"$variavel_php - 150\" e agora vale " +
variavel2_js);
alert("Para receber o valor da variável do PHP no navegador, a página já deve
ter vindo do servidor.");
document.write("variavel2_js recebeu \"$variavel_php - 150\" e agora vale " +
variavel2_js);
document.write("<br>");
</script>

<h2>PHP recebendo de JavaScript</h2>

<?php

$variavel2_php="<script>document.write(variavel2_js)</script>";

print "Usando PHP: A variavel2_php recebeu variavel2_js e agora vale
$variavel2_php<br><br>";

print "Isso acima está errado, pois apenas é algo que tem valor quando chega ao
navegador.<br><br>";

print "Para que o PHP receba de fato o valor de uma variável em JS, esta
variável deve ser recebida por uma variável
em PHP submetida para o servidor, por exemplo o campo de um formulário<br><br>";

// Não podemos concatenar assim: <script>alert('A variável variavel2_php agora
vale' + $variavel2_php)</script>
// Pois a variável $variavel2_php contém <script>alert(...

print "<script>document.write('Usando JS: A variável variavel2_php agora vale' )
</script> $variavel2_php";

?>
<script>var varSigla="DNOCS";</script>
```

```
<form method="POST" action="php_js2.php">
<br><br>
Clique Neste campo e mude o foco:
<input name="sigla" maxlength="13" value="ValorOriginal"
onFocus="this.value=varSigla">
<input type=submit value=Enviar>
</form>

<?php

print "<font color='red'><h4>Para Ajuda com a depuração do JavaScript:
usando o Firefox vá em Ferramentas - Console de Erros</h4></font>";

echo "<hr><h2>Código Fonte</h2>";

show_source(__FILE__);
?>
```

```
<?php
// Arquivo php2_js.php
if ($_POST['sigla']){
        echo "A variável sigla, campo do Form que valia 'ValorOriginal', recebeu
o valor da variável JS 'DNOCS'<br>";
        echo "que foi atribuído à variável \$sigla do PHP através do campo do
form!";
}

echo "<hr><h2>Código Fonte</h2>";

show_source(__FILE__);

?>
```

Outro Exemplo de Integração do PHP com JavaScript

```
<h1>Integração entre PHP e JavaScript</h1>

<script>
var nome;
nome = prompt('Qual o seu nome?','João Brito Cunha');
</script>

<h2>PHP Mostrando variável JavaScript</h2>
<?php
echo "Seu nome é <script>document.write(nome)</script>";
echo "<script>alert('Seu nome é '+nome)</script>";
?>

<h2>JavaScript Mostrando variável PHP</h2>

<?php
$arquivo="phpjs.php";
$acao='excluir';
$tipo='diretorio';
```

```
echo "<br><br>Arquivo = $arquivo  Ação = $acao e Tipo = $tipo";
?>

<script>

if(!confirm("Confirma?")){
    alert("Não confirmou!");
}else{
    alert("Arquivo = <?=$arquivo?> Acao = <?=$acao?> Tipo = <?=$tipo?>");
}
</script>

<h2>HTML e JavaScript Mostrando variável PHP</h2>

<?php $sigla="DNOCS"; ?>

<form>
Clique Neste campo e mude o foco<br>
<input name="sigla" maxlength="10" value="<?=$sigla?>"
        onBlur="javascript:location.href='<?=$PHP_SELF?>?sigla=document.forms[0]
.sigla.value'">
</form>
```

PHP com banco de dados Access

http://www.imasters.com.br/artigo/299/php/php_com_banco_de_dados_access/

Sugestão de Livros

```
Desenvolvendo Web Sites com PHP
Editora Novatec
Juliano Niederauer

PHP para quem conhece PHP
Editora Novatec
Juliano Niederauer

PHP Guia do Desenvolvedor
Ed. Berkeley
Sterling Hughes
(Do grupo de desenvolvimento do PHP)

Desvendando aplicações na Web com PHP 4.0
Ed. Ciência Moderna
Tobias Retschiller e
Till Gerken
```

Links

PHP

http://www.php.net

http://www.php.net/manual/pt_BR/

http://www.php.net/downloads.php

http://www.phpbrasil.com/

http://www.planet-source-code.com/vb/default.asp?lngWId=8

http://www.hotscripts.com/PHP/index.html

http://www.zend.com (empresa dos dois membros israelenses da equipe de desenvolvimento do PHP, especial ênfase na seção Developer Zone)

http://www.phpwizard.net

http://www.phpclasses.org

http://www.weberdev.com

http://www.devshed.com

http://www.phpmania.org/

http://www.phpnet.us/ - Hospedagem free para PHP e MySQL (300MB)

http://www.superphp.com.br

http://www.scriptbrasil.com/

http://www.faqts.com/knowledge_base/index.phtml/fid/51/

http://www.alt-php-faq.org/

http://ribafs.byethost2.com/

CURSOS GRÁTIS

http://cursos.cdtc.org.br/brasil/ - Cursos Grátis para Func.Públ.

http://www.apostilando.com/download.php?cod=171&categoria=PHP

http://www.aprendaemcasa.com.br/apcasa6.htm

http://www.solocursosgratis.com/cursos_gratis_php-slctema264.htm

CLIPARTS http://www.digitmania.holowww.com/digital.html – Dígitos (cliparts)

POSTGRESQL

http://www.postgresql.org/docs/current/interactive/

http://pgdocptbr.sourceforge.net/pg80/index.html

http://www.designmagick.com/category/3/PostgreSQL

MYSQL

http://dev.mysql.com/doc/refman/4.1/pt/index.html

http://dev.mysql.com/doc/refman/5.0/en/index.html

SQL

http://www.firstsql.com/tutor.htm

http://sqlzoo.net/

http://www.sql-tutorial.net/SQL-tutorial.asp

http://www.programmingtutorials.com/sql.aspx

http://tutorials.findtutorials.com/

http://www.hardened-php.net/home.8.html - Hardened PHP Project

http://www.1phpscripts.com/

http://www.weberdev.com/

http://www.goldsofts.com/scriptscategory/10/0/1/0.html

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/PHP/PHP2"
Page categories: PHP

Aplicativos em PHP/Referências/E-books free

De Wikibooks

< Aplicativos em PHP | Referências

Conteúdo

- 1 Livros e E-Books
- 2 E-Books Free
  - 2.1 PHP 5 Power Programming
  - 2.2 Muitos E-Books sobre diversas áreas
  - 2.3 Projeto Gutemberg com mais de 18.000 e-books frees
- 3 Livros
  - 3.1 Iniciante
  - 3.2 Avançado

Livros e E-Books

E-Books Free

PHP 5 Power Programming

Livro muito bom sobre PHP5 - http://www.phptr.com/content/images/013147149X/downloads/013147149X_book.pdf

Muitos E-Books sobre diversas áreas

- http://www.e-book.com.au/freebooks.htm
- http://www.intelligentedu.com/free_computer_books.html

Projeto Gutemberg com mais de 18.000 e-books frees

http://www.gutenberg.org/index.php

Livros

Iniciante

Desenvolvendo Websites com PHP

Juliano Niederauer

Páginas: 272

Avançado

PHP para quem conhece PHP

Juliano Niederauer

Páginas: 480

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/E-books_frees"
Page categories: PHP

Aplicativos em PHP/Referências/Cursos na área

De Wikibooks

< Aplicativos em PHP | Referências

Conteúdo

- 1 Cursos na Área
- 2 Cursos Gratuitos
  - 2.1 Cursos Gratuitos para Funcionários Públicos
  - 2.2 Cursos Gratuitos abertos para a Comunidade
  - 2.3 Cursos de Linux Online Grátis
  - 2.4 Curso de HTML, PHP e outros
  - 2.5 Curso de JavaScript
  - 2.6 Cursos de PHP Orientado a Objetos e outros
- 3 Cursos Pagos

Cursos na Área

Cursos Gratuitos

Cursos Gratuitos para Funcionários Públicos

Cursos de Apache, PHP, Joomla, Xoops, Moodle, MySQL, PostgreSQL, HTML, etc. Todos de ótima qualidade e com direito a certificado -

http://cursos.cdtc.org.br/brasil/

Cursos Gratuitos abertos para a Comunidade

A mesma estrutura oferecendo os mesmos cursos abertos à comunidade

http://comunidade.cdtc.org.br/

Cursos de Linux Online Grátis

Cursos de:

SGBDs - MySQL, Oracle e PostgreSQL

Hardware

Internet

Linux - Básico, Intermediário e Avançado

Open-Office - Writer, Calc e Impress

Programação - C, C++, Java, JSP, Perl, PHP e ShellScript

Redes

WebDesign - BlueFish-NVU, DreamWeaver, Fhash, Gimp, HTML-CSS, Inkscape, JavaScript, OpenOffice-Draw e PhotoShop

http://www.linuxbrasil.org.br/

Curso de HTML, PHP e outros

http://portalfarmaceuticon.com/cursos/index.php

Curso de JavaScript

http://neosite.ilogic.com.br/dicas/2007/04/curso-gratuito-programando-em-javascript.html

Cursos de PHP Orientado a Objetos e outros

http://cursos.ribafs.net

Cursos Pagos

UNIFOR nas Férias - http://www.unifor.br (Fortaleza - Ce - Brasil)

Evolução - http://www.evolucao.com.br/ (Fortaleza - Ce - Brasil)

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Refer%C3%AAncias/Cursos_na_%C3%A1rea"
Page categories: PHP

Aplicativos em PHP/Administração dos SGBDs/MySQL

De Wikibooks

< Aplicativos em PHP | Administração dos SGBDs

Conteúdo

- 1 Instalação
- 2 Executando
- 3 Trocando a senha do usuário root
- 4 Criando Usuários e Concedendo Privilégios
- 5 Removendo Usuários
- 6 Instalando como Serviço
- 7 Criando Bancos e Tabelas
- 8 Usando o MySQL
- 9 Importação e Exportação de Dados e Extrutura
- 10 Mudar Conjunto de Caracters para LATIN1
- 11 Importar CSV no MySQL
- 12 Funções com Datas
- 13 Ativando o suporte a INNODB no MySQL do XAMPP
- 14 Conversão de Funções do MySQL para o PostgreSQL no PHP
- 15 Replicação no MySQL
- 16 Migrando .DBF para MySQL
- 17 phpMyAdmin
- 18 Conversão de Tipos
  - 18.1 Convertendo varchar em date no MySQL
- 19 Referências
- 20 Como o MySQL pode facilitar a sua vida

Instalação

Para testes locais uma instalação prática é com o pacote Xampp

http://xampp.sf.net

Para uso em servidores onde se exige segurança idealmente instalar compilando os fontes para um comtrole maior das características instaladas.

Executando

Uma boa opção de administração do MySQL é o phpmyadmin, que também acompanha o Xampp.

Para administração pela linha de comando use:

mysql -h host -u user -p (o super usuário default é root)

mysql -u root (quando estiver sem senha)

Trocando a senha do usuário root

```
Acessar o servidor do MySQL:
mysql -u root mysql (Usuário root acessar banco mysql)

Alterar senha atual do root para novasenha:
UPDATE user SET Password=PASSWORD("novasenha") WHERE user="root";

Atualizar os procedimentos:
FLUSH PRIVILEGES;
```

Criando Usuários e Concedendo Privilégios

mysql --user=root mysql

GRANT ALL PRIVILEGES ON *.* TO super@localhost IDENTIFIED BY 'senha' WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON *.* TO super@"%" IDENTIFIED BY 'some_pass' WITH GRANT OPTION;

super - é um total super usuário que pode se conectar no localhost e de qualquer lugar ("%"), mas precisa usar senha

GRANT RELOAD,PROCESS ON *.* TO admin@localhost;

admin - usuário que pode se conectar no localhost sem senha.

Pode executar os comandos mysqladmin reload, mysqladmin refresh, and mysqladmin flush-* e mysqladmin processlist . Não tem nenhum privilégio relacionado aos bancos.

GRANT USAGE ON *.* TO fraco@localhost;

fraco - pode conectar somente via localhost sem senha mas sem privilégios, somente para uso.

Exemplo:

GRANT ALL PRIVILEGES ON *.* TO ribafs@localhost IDENTIFIED BY 'ribafs' WITH GRANT OPTION;

mysql -u ribafs // Dá erro de senha

mysql -u ribafs -p //Funciona após entrar a senha ribafs

Criando Usuários de Olho na Segurança

Este usuário somente conecta o MySQL via localhost.

```
- Abrir phpMyAdmin como super usuário
- Clicar no link Privilégios
- Clicar em Adicionr novo usuário
- Entre com o nome do usuário
- Em servidor selecione Local
- Em senha selecione Sem senha
- Abaixo em Privilégios globais selecione todas as checkbox Dados
- Caso queira marcar mais algum dos privilégios, faça-o e clique em Executar.
```

Pronto, este usuário somente poderá realizar conexões locais e não terá nenhum privilégio a não ser os de cadastrar os dados (nada de excluir, criar ou alterar a estrutura do banco).

Removendo Usuários

DROP USER nomeusuario;

Privilégios

REVOKE GRANT ALL ON nomebancooutabelaou*ou*.* FROM nomeusuario

- - todas as tabelas
- .* todos os bancos e todas as tabelas

banco.* - todas as tabelas do banco

GRANT SELECT,INSERT,UPDATE ON nomebanco.* TO nomeuser;

GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON nomebanco.* TO usuario@localhost IDENTIFIED BY 'senha';

GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON nomebanco.* TO usuario@dominio.com.br IDENTIFIED BY 'senha';

GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON nomebanco.* TO usuario@'%' IDENTIFIED BY 'senha';

Instalando como Serviço

Instalar MySQL como serviço no Windows para trabalhar com Java (J2EE):

mysqld-nt --install --ansi --sql-mode=ANSI_QUOTES

Instalar como serviço:

bin\mysqld-nt --install mysql

Remover o serviço:

bin\mysqld --remove mysql

Remover serviço ansi:

bin\mysqld --remove --ansi

Removendo Serviços no Windows XP/NT

mysql\bin\mysqld -- remove (remove o serviço mysql) -- remove --ansi (remover o serviço ansi)

Criando Bancos e Tabelas

```
CREATE DATABASE db_name
[[DEFAULT] CHARACTER SET charset_name]
[[DEFAULT] COLLATE collation_name]
ALTER DATABASE db_name
[[DEFAULT] CHARACTER SET charset_name]
[[DEFAULT] COLLATE collation_name]

CREATE TABLE tbl_name (column_list)
[[DEFAULT] CHARACTER SET charset_name] [COLLATE collation_name]]
ALTER TABLE tbl_name
[[DEFAULT] CHARACTER SET charset_name] [COLLATE collation_name]
Example:
CREATE TABLE t1 ( ... ) CHARACTER SET latin1 COLLATE latin1_danish_ci;

col_name {CHAR | VARCHAR | TEXT} (col_length)
[CHARACTER SET charset_name] [COLLATE collation_name]

Exemplo:
CREATE TABLE Table1
(
column1 VARCHAR(5) CHARACTER SET latin1 COLLATE latin1_german1_ci
);
```

Criar Tabelas com Relacionamentos

```
create table produto(
        codigo int not null primary key,
        nome varchar(50) not null unique,
        descricao varchar(200),
        valor real(6, 2)
) ENGINE=INNODB;

create table cliente(
        codigo int not null primary key,
        nome varchar(50) not null,
        email varchar(100) not null unique,
        cpf varchar(11) not null
) ENGINE=INNODB;

create table pedido(
        numero int not null primary key auto_increment,
        codigocliente int not null references cliente(codigo),
        valortotal real(7,2) DEFAULT '0.00' NOT NULL
) ENGINE=INNODB;

create table item(
        numeropedido int not null references pedido(numero),
        codigoproduto int not null references produto(codigo),
        quantidade int not null,
        primary key(numeropedido, codigoproduto)
) ENGINE=INNODB;

CREATE TABLE product (
        category INT NOT NULL,
        id INT NOT NULL,
        price DECIMAL,
        PRIMARY KEY(category, id)
) ENGINE=INNODB;

CREATE TABLE product_order (
        no INT NOT NULL AUTO_INCREMENT,
        product_category INT NOT NULL,
        product_id INT NOT NULL,
        customer_id INT NOT NULL,
        PRIMARY KEY(no),
        INDEX (product_category, product_id),
        FOREIGN KEY (product_category, product_id) REFERENCES product(category,
id)
           ON UPDATE CASCADE ON DELETE RESTRICT,
        INDEX (customer_id),
        FOREIGN KEY (customer_id) REFERENCES customer(id)
) ENGINE=INNODB;
```

O tipo InnoDb dá suporte à constraint Foreign Key (references).

Usando o MySQL

1) mysql -u root -p ou mysql -u root

mysql -h host -u user -p banco

Obs: Caso receba a mensagem: Can't connect to MySQL server on 'localhost'

Falta startar o MySQL

2) create database nomebanco;

3) use nomebanco;

4) create table nometabela(campos tipos...);

5) select * from nometabela;

6) show databases;

7) show tables;

8) describe nometabela;

Importação e Exportação de Dados e Extrutura

Exportando:

bin\mysqldump -u user -p passwd banco > banco.sql

Importando:

bin\mysql -u user -p password banco < banco.sql

IMPORTAR Todos os Bancos de um Script

Temos um script contendo diversos bancos então:

mysql -u root < varios_bancos.sql

Mudar Conjunto de Caracters para LATIN1

musql -u root

\C latin1

Importação e Exportação com o phpMyAdmin

Exportar todo um banco

- Abrir o phpMyAdmin e selecionar o banco
- Clicar no botão Exportar

- Basta rolar a tela e clicar em Executar abaixo

- O conteúdo será exibido na tela. Devemos então selecionar e criar um arquivo texto com o mesmo. Por convenção criamos arquivos com a extensão .sql.

- Também podemos exportar para um arquivo compactado. Basta clicar na opção "Compactado com zip ou gzip.

- Existem muitas outras opções para a exportação.

Caso queira exportar somente uma tabela o processo é o mesmo, mudando apenas que devemos selecionar apenas a tabela desejada.

Populando Tabelas após a criação

O comando LOAD DATA pode ser utilizado para popular tabelas, trazendo de arquivos:

LOAD DATA LOCAL INFILE '/path/arquivo.txt' INTO TABLE nometabela;

SELECT DATABASE();

SHOW CHARACTER SET;

Importar CSV no MySQL

```
$handle = fopen ('./file.csv', 'r');
                while (($data = fgetcsv($handle, 1000, ',', '"')) !== FALSE)
                {
                        $query = "INSERT INTO services VALUES ('".
implode("','", $data)."')";
                        $query = @mysql_query($query);
                }


LOAD DATA INFILE "./ImportData.csv"
     INTO TABLE table1
     FIELDS TERMINATED BY ","
     OPTIONALLY ENCLOSED BY """"
     LINES TERMINATED BY "\r\n";

OPTIONALLY ENCLOSED is optional.
```

Funções com Datas

```
DATE_SUB
SELECT something FROM tbl_name WHERE DATE_SUB(CURDATE(),INTERVAL 30 DAY) <=
date_col;
SELECT DATEDIFF('1997-12-31 23:59:59','1997-12-30');

DATE_ADD
SELECT DATE_ADD('2006-05-00',INTERVAL 1 DAY);

SELECT CURDATE();
```

```
SELECT CURTIME();

DATE_FORMAT
SELECT date_format( '2006-04-30', '%d/%m/%Y' ); -- 30/04/2006
SELECT DATE_FORMAT('2003-10-03',GET_FORMAT(DATE,'EUR')); -- 03.10.2003
SELECT DATE_FORMAT('2006-06-00', '%d/%m/%Y');

SELECT NOW();

SELECT TO_DAYS('1997-10-07'); -- RETORNA DIAS

SELECT YEAR('2000-01-01');
```

Ativando o suporte a INNODB no MySQL do XAMPP

A versão atual do phpMyAdmin que vem no Xampp 1.62 ainda vem sem o suporte a InnoDB.

Acontece que se você baixar a versão do phpMyAdmin atual (2.10.2) esta já vem com o suporte ativado e este tutorial não se faz necessário, apenas para versões anteriores. Acredito que versões posteriores do Xampp também já venham com esse suporte ativado.

Detalhe: Apenas faça o download do phpMyAdmin (http://www.phpmyadmin.net), descompacte no diretório web e sem nenhuma configuração abra no navegador.

O tipo de tabelas padrão do MySQL (MyISAM) não oferece suporte a relacionamentos (chave estrangeira). Para isso precisamos ativar o suporte a um tipo de tabelas de terceitos (InnoDB).

Editar o arquivo my.cnf:

D:_xampplite\mysql\bin\my.cnf

Caso apareça um arquivo "my" sem extensão clique com o botão direito, enviar para e abra num editor de texto e faça as seguintes alterações:

Comentar a linha:

#skip-innodb

Descomentar as linhas:

```
innodb_data_home_dir = D:/_xampplite/mysql/data/
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = D:/_xampplite/mysql/data/
innodb_log_arch_dir = D:/_xampplite/mysql/data/

set-variable = innodb_buffer_pool_size=16M
set-variable = innodb_additional_mem_pool_size=2M

set-variable = innodb_log_file_size=5M
set-variable = innodb_log_buffer_size=8M
innodb_flush_log_at_trx_commit=1
set-variable = innodb_lock_wait_timeout=50
```

Pronto. Reinicie o MySQL e agora você pode criar tabelas com suporte a INNODB (consequentemente foreign key e relacionamentos).

Conversão de Funções do MySQL para o PostgreSQL no PHP

Na conversão de scripts em PHP com MySQL para PHP com PostgreSQL temos que atendar para vários detalhes: nomes das funções (ex.: mysql_connect para pg_connect), sintaxe das funções, que varia em muitas delas (veremos adiante), estrutura do banco: tipos de tabelas, tipos de dados, auto-incremento, etc. Óbvio que para converter entre dois SGBDs devemos conhecer as características de ambos. Exemplo: como vou usar um tipo de dados do MySQL no PostgreSQL, se este não tem o referido tipo? Deverei encontrar no PostgreSQL, um tipo que satisfaça os requisitos daquele do MySQL (para isso precisarei conhecer as características dos tipos de ambos, suas faixas de valores, tipos de dados, etc).

Alguns Exemplos de Conversão de Funções. Não vou me preocupar com tratamento de erros nem com outros detalhes, mostrarei apenas as funções para comparar:

Conexão ao Banco de Dados

No MySQL:

Abrir a conexão:

$con_my = mysql_connect("localhost:porta", "usuario", "senha");

Quando for usar, selecionar o banco:

mysql_select_db('nomebanco', $con_my);

No PostgreSQL:

Etapa única:

$con_pg = pg_connect("host=127.0.0.1 port=5432 dbname=banco user=usuario password=senha");

Obs.: na conexão já se seleciona o banco a ser usado.

Consultas

MySQL:

$q = mysql_query(“SELECT * FROM tabela”, $con_my);

PostgreSQL:

$q = pg_query($con_pg, “SELECT * FROM tabela”);

Obs.: Veja que a ordem dos parâmetros é invertida.

LIMIT

Aqui temos uma boa diferença entre ambos.

Quando for a forma resumida não há diferença entre ambos. Por exemplo:

SELECT * FROM tabela ORDER BY campo LIMIT 5;

Obs.: Altamente recomendado usar ORDER BY antes de LIMIT, para um retorno coerente e o campo do ORDER BY deve ser o campo chave primparia.

Quando o LIMIT trouxer os dois parâmetros então teremos diferença.

Sintaxe no PostgreSQL:

SELECT lista_de_campos FROM expressão [LIMIT { número | ALL }] [OFFSET inicio]

LIMIT ALL – mesmo que omitir LIMIT.

OFFSET inicio – orienta para que a consulta retorne somente a partir de inicio.

OFFSET 0 – mesmo que omitir OFFSET.

LIMIT 50 OFFSET 11 – Deverá trazer 50 registros do 12 até o 50, caso existam.

Exemplos:

SELECT * FROM cliente ORDER BY codigo LIMIT 2 OFFSET 0;

Irá retornar os registros do 1 e 2.

SELECT * FROM cliente ORDER BY codigo LIMIT 2 OFFSET 1;

Irá retornar os registros do 2 e 3.

SELECT * FROM cliente ORDER BY codigo LIMIT 2 OFFSET 2;

Irá retornar os registros do 3 e 4 (Se existirem).

Ou seja, o primeiro parâmetro do é a quantidade e o segundo o inicial (começando do 0).

Sintaxe no MySQL:

LIMIT [início,] linhas

Retorna o número de linhas especificado. Se o valor início for fornecido, aquelas linhas são puladas antes do dado ser retornado. A primeira linha é 0.

Exemplo:

SELECT * FROM cliente ORDER BY codigo LIMIT 3,2;

O comando acima pede apenas os registros de código 4 e 5, os dois após o 3.

SELECT * FROM cliente ORDER BY codigo LIMIT 3,1;

Aqui retornará 1 registro, de código 4, que é o próximo após o 3.

SELECT * FROM cliente ORDER BY codigo LIMIT 2 , 4;

```
    Aqui retornará os registros de código 3,4,5 e 6.
```

No MySQL o início é o primeiro parâmetro e a quantidade é o segundo.

Experimente os exemplos acima num gerenciador dos SGBDs para consolidar o conhecimento (phpmyadmin e phppgadmin).

Replicação no MySQL

An introduction to replication1

How to start replicating - the slave server

Migrando .DBF para MySQL

No ótimo site Vivaolinux - http://www.vivaolinux.com.br/dicas/verDica.php?codigo=8792

phpMyAdmin

Software para administração web do MySQL, que conta com muitos recursos e tem interface simples de usar.

Site oficial - http://www.phpmyadmin.net/home_page/index.php

Download - http://www.phpmyadmin.net/home_page/downloads.php

Documentação principal - http://www.phpmyadmin.net/documentation/

FAQs - http://www.phpmyadmin.net/documentation/#faq

Doing More With phpMyAdmin: part 1 - http://www.devshed.com/c/a/PHP/Doing-More-With-phpMyAdmin-Part-1/

Doing More With phpMyAdmin: part 2 - http://www.devshed.com/c/a/PHP/Doing-More-With-phpMyAdmin-Part-2/

Conversão de Tipos

Convertendo varchar em date no MySQL

Autor: Fábio Berbert de Paula <fberbert@gmail.com> Data: 23/07/2007

Convertendo varchar em date no MySQL

Peguei uma manutenção de um sistema PHP/MySQL pra fazer e surgiu a necessidade de ordenar determinada tabela pelo campo data, porém pra minha surpresa esse campo era do tipo VARCHAR() ao invés de DATE() ou DATETIME().

A data estava armazenada na tabela no formato "dd/mm/yyyy", exemplo:

SELECT data FROM contas;

```
+------------+
| data       |
+------------+
| 26/11/2003 |
| 04/12/2003 |
| 28/11/2003 |
| 05/12/2003 |
| 29/12/2003 |
+------------+
```

O problema é que ao mandar ordenar a tabela por data, por ser VARCHAR o resultado não funciona como o esperado:

SELECT data FROM contas ORDER BY data;

```
+------------+
| data       |
+------------+
| 04/12/2003 |
| 05/12/2003 |
| 26/11/2003 |
| 28/11/2003 |
| 29/12/2003 |
+------------+
```

Isso acontece porque se o campo é texto, ele começa a ordenar da esquerda pra direita em ordem alfanumérica, onde 04/12 é menor que 26/11, o que está errado, visto que em data a gente precisa levar em conta, em ordem de prioridade, ano-mês-dia.

Pra resolver esse problema sem ter de mexer na tabela (o que não tinha permissão pra fazer), use a função str_to_date() do MySQL:

SELECT str_to_date(data, '%d/%m/%Y') AS data FROM contas ORDER BY data;

```
+------------+
| data       |
+------------+
| 2003-11-26 |
| 2003-11-28 |
| 2003-12-04 |
| 2003-12-05 |
| 2003-12-29 |
+------------+
```

Hmmm, agora sim! A sintaxe da função é:

str_to_date(CAMPO, 'formato armazenado da string')

Maiores informações:

* http://dev.mysql.com/...#function_str-to-date

Fonte: http://www.vivaolinux.com.br/dicas/verDica.php?codigo=9000

Referências

- Manual Online do MySQL 4.1 em Português - com opção de busca pelo conteúdo.

http://dev.mysql.com/doc/refman/4.1/pt/index.html

- Manual do MySQL em vários idiomas e em vários formatos para download - http://mysql.org/doc/

- Removendo Duplicidades em MySQL - http://www.dicas-l.com.br/print/20060930.html

Como o MySQL pode facilitar a sua vida

Artigo do Diego Hellas no PHPBrasil

http://phpbrasil.com/articles/article.php/id/1361

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Administra%C3%A7%C3%A3o_dos_SGBDs/MySQL"
Page categories: PHP

Aplicativos em PHP/Administração dos SGBDs/PostgreSQL

Conteúdo

- 1 PostgreSQL Prático
- 2 Tutorial sobre o phpPgAdmin
- 3 Criando Chave Estrangeira Composta no phpPgAdmin e no PGAdmin
  - 3.1 phpPgAdmin
  - 3.2 PGAdmin
- 4 PlPgSQL Tutorial
  - 4.1 Delimitadores
  - 4.2 Declaração de Variáveis
  - 4.3 Declarado Funções
  - 4.4 Alias para Parâmetros de Funções
  - 4.5 Executando Funções
  - 4.6 Tipos Polimórficos
  - 4.7 Copiando Tipos
  - 4.8 Tipos row
  - 4.9 Tipo registro (record)
  - 4.10 Renomeando Variáveis
  - 4.11 Atribuições
  - 4.12 Execução de Expressão ou Consulta sem Resultado
  - 4.13 Execução de Comandos Dinâmicos
  - 4.14 Estruturas de Controle
  - 4.15 Condicionais
  - 4.16 Laços através do resultado de consultas
  - 4.17 Capturar Erros
  - 4.18 Declaração de Variáveis do tipo Cursor
  - 4.19 Erros e Mensagens
  - 4.20 Procedimentos de Gatilho (Trigger)
  - 4.21 NEW
  - 4.22 OLD
  - 4.23 TG_NAME
  - 4.24 TG_WHEN
  - 4.25 TG_LEVEL
  - 4.26 TG_OP
  - 4.27 TG_RELID
  - 4.28 TG_RELNAME
  - 4.29 TG_NARGS
  - 4.30 TG_ARGV[]
  - 4.31 Gatilho para registrar inserções e atualizações
  - 4.32 Gatilho para auditoria
  - 4.33 Autoditoria ao nível de campos
  - 4.34 Gatilho para manter uma tabela sumário

PostgreSQL Prático

http://pt.wikibooks.org/wiki/PostgreSQL_Pr%C3%A1tico

Com isso evitando repetições ou a reinvenção da roda.

Tutorial sobre o phpPgAdmin

- Baixar de - http://phppgadmin.sourceforge.net/

- Copiar para o diretório web e descompactar no diretório "phppgadmin"

- Editar o arquivo conf/config.inc.php e alterar para dois servidores (um local e outro remoto) caso assim queira:

```
        //...
        // Servidor Local
        // Display name for the server on the login screen
        $conf['servers'][0]['desc'] = 'Local';


        // Hostname or IP address for server.  Use '' for UNIX domain socket.
        // use 'localhost' for TCP/IP connection on this computer
        $conf['servers'][0]['host'] = '127.0.0.1';
        $conf['servers'][0]['defaultdb'] = 'nomebancodefault';

        //...
        // Servidor Remoto
        // Example for a second server
        $conf['servers'][1]['desc'] = 'Remoto';
        $conf['servers'][1]['host'] = '10.99.00.11';
        $conf['servers'][1]['port'] = 5432;
        $conf['servers'][1]['defaultdb'] = 'nomebancodefault';

        //...
        // If extra login security is true, then logins via phpPgAdmin with no
        // password or certain usernames (pgsql, postgres, root, administrator)
        // will be denied. Only set this false once you have read the FAQ and
        // understand how to change PostgreSQL's pg_hba.conf to enable
        // passworded local connections.
        $conf['extra_login_security'] = false;

        // Importantes para Importar/Exportar
        $conf['servers'][0]['pg_dump_path'] = '/usr/local/pgsql/bin/pg_dump';
        $conf['servers'][0]['pg_dumpall_path'] =
'/usr/local/pgsql/bin/pg_dumpall';
```

- Pronto, basta abrir http://localhost/phppgadmin

Criando Chave Estrangeira Composta no phpPgAdmin e no PGAdmin

O exemplo é para uma chave com 4 campos, mas o procedimento é válido para qualquer chave composta.

Relacionar composicoes com precos.

Quero que ao digitar um preço em composições o SGBD verifique a integridade do mesmo em precos.

Temos uma tabela cuja chave primária é composta por 4 campos:

tabela, insumo_grupo, insumo, fornecedor e queremos relacionar com outra tabela que também contém os mesmos campos com os mesmos tipos de dados.

Como isso pode ter algum engano, seguem os passos.

Esquema das tabelas:

```
CREATE TABLE "composicoes" (
    "servico" character varying(10) NOT NULL,
    "insumo_grupo" character varying(6) NOT NULL,
    "insumo" character varying(10) NOT NULL,
    "quantidade" real NOT NULL DEFAULT 0,
    "coeficiente" real NOT NULL DEFAULT 0,
    "data_inclusao" timestamp without time zone,
    "data_alteracao" timestamp without time zone,
    "data_cancelamento" timestamp without time zone,
    "uid_inclusao" character varying(10),
    "uid_alteracao" character varying(10),
    "uid_cancelamento" character varying(10),
    "tabela" smallint NOT NULL DEFAULT 2006,
    "fornecedor" smallint NOT NULL DEFAULT 1,
    CONSTRAINT "composicoes_fk_insumo" FOREIGN KEY (insumo_grupo, insumo)
REFERENCES insumos(grupo, insumo),
    CONSTRAINT "composicoes_fk_servicos" FOREIGN KEY (servico) REFERENCES
servicos(servico),
    CONSTRAINT "composicoes_pkey" PRIMARY KEY (servico, insumo_grupo, insumo)
) WITH OIDS;

CREATE TABLE "precos" (
    "tabela" smallint NOT NULL,
    "insumo_grupo" character varying(6) NOT NULL,
    "insumo" character varying(10) NOT NULL,
    "fornecedor" smallint NOT NULL,
    "custo_produtivo" real,
    "custo_improdutivo" real,
    "data_coleta" timestamp without time zone,
    "data_inclusao" timestamp without time zone,
    "data_alteracao" timestamp without time zone,
    "data_cancelamento" timestamp without time zone,
    "uid_inclusao" character varying(10),
    "uid_alteracao" character varying(10),
    "uid_cancelamento" character varying(10),
    CONSTRAINT "precos_fornecedor_fk" FOREIGN KEY (fornecedor) REFERENCES
cadastro.fornecedor(codigo_fornecedor) ON UPDATE RESTRICT ON DELETE RESTRICT,
    CONSTRAINT "precos_insumos_fk" FOREIGN KEY (insumo_grupo, insumo) REFERENCES
insumos(grupo, insumo),
```

```
    CONSTRAINT "precos_pk" PRIMARY KEY (tabela, insumo_grupo, insumo,
fornecedor)
) WITH OIDS;
```

phpPgAdmin

Observe que na tabela de destino (precos), de onde queremos trazer uma informação, a ordem da chave é "tabela, insumo_grupo, insumo e fornecedor". Já em coposicoes, estes campos estão em ordem diferente. Mas no momento de criar a chave estrangeira devemos adicionar na mesma ordem da precos. Veja abaixo.

- Abrir a tabela (composicoes) para a qual queremos inserir uma chave estrangeira no phpPgAdmin

- Clicar na aba Restrição

- Clicar em adicionar chave estrangeira

- Digitar um nome para a chave: composicoes_precos_fk

- Agora vem a dica: selecione nesta ordem à esquerda e clique na seta dupla após selecionar cada um:

```
 - primeiro tabela

 - depois insumo_grupo

 - depois insumo

 - por fim fornecedor (caso altere esta ordem a chave não será criada)
```

- Em Tabela alvo selecione preco

- Clique em Adicionar (abaixo)

- Á esquerda selecione os quatro campos de precos (tabela, insumo_grupo, insumo e fornecedor) e clique na seta dupla para a direita.

- Clique em Adicionar abaixo.

PGAdmin

- Abrir a tabela no PgAdmin (duplo clique no nome da tabela à esquerda)

- Clique em Constraints

- Selecione Fireign Key abaixo e clique em Add

- Digite um nome em name "composicoes_precos_fk"

- Em references selecione "precos"

- Clique na aba Columns

- Em Local column selecione 'tabela'

- Em Referecing selecione 'tabela' e clique em Add

- De forma semelhante adicione os 3 pares restantes: insumo_grupo - insumo_grupo, insumo - insumo e fornecedor - fornecedor.

Tive problema ao adicionar insumo, então adicionei fornecedor antes e depois insumo e foi bem.

- Então clique em OK, confira e OK novamente.

PlPgSQL Tutorial

Esta é uma das linguagens de funções armazenadas com que trabalha o PostgreSQL. É a mais popular das linguagens no PostgreSQL.

Delimitadores

O código de uma função plpgsql é especificado em CREATE FUNCTION como uma string literal delimitado por aspas.

1 Apóstrofo (corpo da função)

CREATE FUNCTION olamundo() RETURNS integer AS '

. . . .

' LANGUAGE plpgsql;

Inicia após AS e termina antes de LANGUAGE

2 Apóstrofos (mascar string literal no corpo da função)

a_output := *Blah*;

SELECT * FROM users WHERE f_name=*foobar*;

4 Apóstrofos (string constante no corpo da função)

a_output := a_output || *AND name LIKE* ***'foobar'*** *AND xyz*

Mais detalhes na documentação oficial.

Declaração de Variáveis

```
user_id integer;
quantity numeric(5);
url varchar;
myrow tablename%ROWTYPE;
myfield tablename.columnname%TYPE;
arow RECORD;
```

Sintaxe geral da declaração de variáveis:

nome [CONSTANT] tipo [NOT NULL] [{ DEFAULT | := } expressão];

Exemplos:

```
quantity integer DEFAULT 32;
url varchar := 'http://mysite.com';
user_id CONSTANT integer := 10;
```

Declarado Funções

```
CREATE FUNCTION func_escopo() RETURNS integer AS $$
DECLARE
quantidade integer := 30;

BEGIN
RAISE NOTICE 'Aqui a quantidade é %', quantidade; -- A quantidade aqui é 30
quantidade := 50;
--
-- Criar um sub-bloco
--
DECLARE
quantidade integer := 80;
BEGIN
RAISE NOTICE 'Aqui a quantidade é %', quantidade; -- A quantidade aqui é 80
END;
RAISE NOTICE 'Aqui a quantidade é %', quantidade; -- A quantidade aqui é 50
RETURN quantidade;
END;
$$ LANGUAGE plpgsql;

Execute
SELECT func_escopo() as escopo;
```

Obs.: O psql exibe as mensagens disparadas pelo RAISE.

Obs2.: BEGIN e END na plpgsql não agrupa transações, apenas grupos de comandos.

Alias para Parâmetros de Funções

Sintaxe:

nome ALIAS FOR $n;

Exemplos:

```
CREATE FUNCTION vendas_taxa(real) RETURNS real AS $$
DECLARE
    subtotal ALIAS FOR $1;
BEGIN
    RETURN subtotal * 0.06;
END;
$$ LANGUAGE plpgsql;

$1 é um alias para o parâmetro da função que é do tipo real.
```

```
CREATE FUNCTION instr(varchar, integer) RETURNS integer AS $$
DECLARE
    v_string ALIAS FOR $1;
    index ALIAS FOR $2;
BEGIN
    -- Aqui fazemos alguns cálculos usando v_string e index
END;
$$ LANGUAGE plpgsql;
CREATE FUNCTION concat_campos(tab tbl_clientes) RETURNS text AS $$
BEGIN
    RETURN tab.nome || '' '' || tab.email || '' '' || tab.cidade || '' '' ||
tab.estado;
END;
$$ LANGUAGE plpgsql;
```

Também podemos (mais claro) usar os nomes dos parâmetros explicitamente:

```
CREATE FUNCTION sales_tax(subtotal real, OUT tax real) AS $$
BEGIN
    tax := subtotal * 0.06;
END;
$$ LANGUAGE plpgsql;

O parâmetro OUT (output, saída) é mais útil quando retorna múltiplos parâmetros:

CREATE FUNCTION sum_n_product(x int, y int, OUT sum int, OUT prod int) AS $$
BEGIN
    sum := x + y;
    prod := x * y;
END;
$$ LANGUAGE plpgsql
```

Executando Funções

Usa-se o select para executar uma função, como se fosse uma view ou consulta comum.

select sum_n_product(5, 6);

Tipos Polimórficos

Quanto tipos polimórficos (anyelement e anyarray) são usados para declarar de funções, um parâmetro especial ($0) é criado. Este tipo de dados é o atual retorno da função.

É inicializado como NULL e pode ser modificado pela função.

Exemplo

Função que trabalha com qualquer tipo de dados e que suporta o operador +:

```
CREATE FUNCTION add_three_values(v1 anyelement, v2 anyelement, v3 anyelement)
RETURNS anyelement AS $$
DECLARE
    result ALIAS FOR $0;
BEGIN
    result := v1 + v2 + v3;
    RETURN result;
END;
$$ LANGUAGE plpgsql;
```

```
Ou:

CREATE FUNCTION add_three_values(v1 anyelement, v2 anyelement, v3 anyelement,
                                 OUT sum anyelement)
AS $$
BEGIN
    sum := v1 + v2 + v3;
END;
$$ LANGUAGE plpgsql;
```

Veja, que ao usar OUT não há necessidade de retorno (já está implícito).

Copiando Tipos

variavel%TYPE

%TYPE fornece o tipo de dados de uma variável ou de um campo de tabela.

Para declarar uma variável com o mesmo tipo de dado de usuarios.id_usuario deve ser escrito:

id_usuario usuarios.id_usuario%TYPE;

Tipos row

Pode armazenar um registro resultante de um SELECT ou de um FOR.

nome nome_da_tabela%ROWTYPE;

nome nome_do_tipo_composto;

Os campos podem ser acessados com nomevariavel.nomecampo;

Exemplo de uso:

```
CREATE FUNCTION mesclar_campos(t_linha nome_da_tabela) RETURNS text AS $$
DECLARE
t2_linha nome_tabela2%ROWTYPE;
BEGIN
SELECT * INTO t2_linha FROM nome_tabela2 WHERE ... ;
RETURN t_linha.f1 || t2_linha.f3 || t_linha.f5 || t2_linha.f7;
END;
$$ LANGUAGE plpgsql;
SELECT mesclar_campos(t.*) FROM nome_da_tabela t WHERE ... ;
```

Tipo registro (record)

nome record;

Renomeando Variáveis

RENAME nome_antigo TO novo_nome;

```
CREATE FUNCTION logfunc1(logtxt text) RETURNS timestamp AS $$
BEGIN
INSERT INTO logtable VALUES (logtxt, 'now');
RETURN 'now';
```

```
END;
$$ LANGUAGE plpgsql;
e
CREATE FUNCTION logfunc2(logtxt text) RETURNS timestamp AS $$
DECLARE
curtime timestamp;
BEGIN
curtime := 'now';
INSERT INTO logtable VALUES (logtxt, curtime);
RETURN curtime;
END;
$$ LANGUAGE plpgsql;
```

Atribuições

identificador := expressão;

```
SELECT INTO meu_registro * FROM emp WHERE nome_emp = meu_nome;
IF NOT FOUND THEN
RAISE EXCEPTION ''não foi encontrado o empregado %!'', meu_nome;
END IF;
```

Execução de Expressão ou Consulta sem Resultado

PERFORM create_mv('cs_session_page_requests_mv', my_query);

Não Fazer Nada

NULL;

Por exemplo, os dois fragmentos de código a seguir são equivalentes:

```
BEGIN
        y := x / 0;
EXCEPTION
        WHEN division_by_zero THEN
        NULL; -- ignorar o erro
END;
BEGIN
        y := x / 0;
EXCEPTION
        WHEN division_by_zero THEN -- ignorar o erro
END;
```

Execução de Comandos Dinâmicos

EXECUTE cadeia_de_caracteres_do_comando;

```
EXECUTE 'UPDATE tbl SET '
|| quote_ident(nome_da_coluna)
|| ' = '
|| quote_literal(novo_valor)
|| ' WHERE key = '
|| quote_literal(valor_chave);

EXECUTE 'UPDATE tbl SET '
|| quote_ident(nome_da_coluna)
```

```
|| ' = $$'
|| novo_valor
|| '$$ WHERE key = '
|| quote_literal(valor_chave);
```

Estruturas de Controle

Return next

RETURN NEXT expressão;

Quando uma função PL/pgSQL é declarada como retornando SETOF algum_tipo, o procedimento a ser seguido é um pouco diferente. Neste caso, os itens individuais a serem retornados são especificados em comandos RETURN NEXT, e um comando RETURN final, sem nenhum argumento, é utilizado para indicar que a função chegou ao fim de sua execução. O comando RETURN NEXT pode ser utilizado tanto com tipos de dado escalares quanto compostos; no último caso toda uma "tabela" de resultados é retornada.

As funções que utilizam RETURN NEXT devem ser chamadas da seguinte maneira:

SELECT * FROM alguma_função();

Condicionais

```
• IF ... THEN
• IF ... THEN ... ELSE
• IF ... THEN ... ELSE IF
• IF ... THEN ... ELSIF ... THEN ... ELSE
• IF ... THEN ... ELSEIF ... THEN ... ELSE

IF linha_demo.sexo = 'm' THEN
sexo_extenso := 'masculino';
ELSE
IF linha_demo.sexo = 'f' THEN
sexo_extenso := 'feminino';
END IF;
END IF;

IF expressão_booleana THEN
instruções
[ ELSIF expressão_booleana THEN
instruções
[ ELSIF expressão_booleana THEN
instruções
...]]
[ ELSE
instruções ]
END IF;

IF numero = 0 THEN
resultado := 'zero';
ELSIF numero > 0 THEN
resultado := 'positivo';
ELSIF numero < 0 THEN
resultado := 'negativo';
ELSE
-- hmm, a única outra possibilidade é que o número seja nulo
```

```
resultado := 'NULL';
END IF;
```

Laços

```
[<<rótulo>>]
LOOP
instruções
END LOOP;
```

Exit

```
EXIT [ rótulo ] [ WHEN expressão ];
```

```
LOOP
-- algum processamento
IF contador > 0 THEN
EXIT; -- sair do laço
END IF;
END LOOP;
LOOP
-- algum processamento
EXIT WHEN contador > 0; -- mesmo resultado do exemplo acima
END LOOP;
BEGIN
-- algum processamento
IF estoque > 100000 THEN
EXIT; -- causa a saída do bloco BEGIN
END IF;
END;
```

While

```
[<<rótulo>>]
WHILE expressão LOOP
instruções
END LOOP;
```

```
WHILE quantia_devida > 0 AND saldo_do_certificado_de_bonus > 0 LOOP
-- algum processamento
END LOOP;
WHILE NOT expressão_booleana LOOP
-- algum processamento
END LOOP;
```

For (laços internos)

```
[<<rótulo>>]
FOR nome IN [ REVERSE ] expressão .. expressão LOOP
instruções
END LOOP;
```

```
FOR i IN 1..10 LOOP
-- algum processamento
RAISE NOTICE 'i é %', i;
END LOOP;
FOR i IN REVERSE 10..1 LOOP
-- algum processamento
END LOOP;
```

Laços através do resultado de consultas

```
[<<rótulo>>]
FOR registro_ou_linha IN comando LOOP
instruções
END LOOP;

CREATE FUNCTION cs_refresh_mviews() RETURNS integer AS $$
DECLARE
        mviews RECORD;
BEGIN
        PERFORM cs_log('Atualização das visões materializadas...');
                FOR mviews IN SELECT * FROM cs_materialized_views ORDER BY
sort_key LOOP
                -- Agora "mviews" possui um registro de cs_materialized_views
                PERFORM cs_log('Atualizando a visão materializada ' ||
quote_ident(mviews.mv_name) || ' ...');
                EXECUTE 'TRUNCATE TABLE ' || quote_ident(mviews.mv_name);
                EXECUTE 'INSERT INTO ' || quote_ident(mviews.mv_name) || ' ' ||
mviews.mv_query;

        END LOOP;
        PERFORM cs_log('Fim da atualização das visões materializadas.');
        RETURN 1;
END;
$$ LANGUAGE plpgsql;

[<<rótulo>>]
FOR registro_ou_linha IN EXECUTE texto_da_expressão LOOP
instruções
END LOOP;
```

Capturar Erros

```
[ <<rótulo>> ]
[ DECLARE
declarações ]
BEGIN
instruções
EXCEPTION
WHEN condição [ OR condição ... ] THEN
instruções_do_tratador
[ WHEN condição [ OR condição ... ] THEN
instruções_do_tratador
... ]
END;


INSERT INTO minha_tabela(nome, sobrenome) VALUES('Tom', 'Jones');
BEGIN
        UPDATE minha_tabela SET nome = 'Joe' WHERE sobrenome = 'Jones';
        x := x + 1;
        y := x / 0;
EXCEPTION
        WHEN division_by_zero THEN
                RAISE NOTICE 'capturado division_by_zero';
                RETURN x;
END;
```

Declaração de Variáveis do tipo Cursor

Todos os acessos aos cursores na linguagem PL/pgSQL são feitos através de variáveis cursor, que sempre são do tipo de dado especial refcursor. Uma forma de criar uma variável cursor é simplesmente declará-la como sendo do tipo refcursor. Outra forma é utilizar a sintaxe de declaração de cursor, cuja forma geral é:

nome CURSOR [(argumentos)] FOR comando ;

Exemplos:

```
DECLARE
curs1 refcursor;
curs2 CURSOR FOR SELECT * FROM tenk1;
curs3 CURSOR (chave integer) IS SELECT * FROM tenk1 WHERE unico1 = chave;

CREATE TABLE teste (col text);
INSERT INTO teste VALUES ('123');
CREATE FUNCTION reffunc(refcursor) RETURNS refcursor AS '
BEGIN
OPEN $1 FOR SELECT col FROM teste;
RETURN $1;
END;
' LANGUAGE plpgsql;
BEGIN;
SELECT reffunc('funccursor');
reffunc
------------
funccursor
(1 linha)
FETCH ALL IN funccursor;

COMMIT;
```

Erros e Mensagens

RAISE nível 'formato' [, variável [, ...]];

Os níveis possíveis são DEBUG, LOG, INFO, NOTICE, WARNING, e EXCEPTION. O nível EXCEPTION causa um erro (que normalmente interrompe a transação corrente); os outros níveis apenas geram mensagens com diferentes níveis de prioridade. Se as mensagens de uma determinada prioridade são informadas ao cliente, escritas no log do servidor, ou as duas coisas, é controlado pelas variáveis de configuração log_min_messages e client_min_messages.

Procedimentos de Gatilho (Trigger)

É criado pelo comando CREATE FUNCTION, declarando o procedimento como uma função sem argumentos e que retorna o tipo trigger. Deve ser observado que a função deve ser declarada sem argumentos, mesmo que espere receber os argumentos especificados no comando CREATE TRIGGER — os argumentos do gatilho são passados através de TG_ARGV, conforme descrito abaixo.

Quando uma função escrita em PL/pgSQL é chamada como um gatilho, diversas variáveis especiais são criadas automaticamente no bloco de nível mais alto. São estas:

NEW

Tipo de dado RECORD; variável contendo a nova linha do banco de dados, para as operações de INSERT/UPDATE nos gatilhos no nível de linha. O valor desta variável é NULL nos gatilhos no nível de instrução.

OLD

Tipo de dado RECORD; variável contendo a antiga linha do banco de dados, para as operações de UPDATE/DELETE nos gatilhos no nível de linha. O valor desta variável é NULL nos gatilhos no nível de instrução.

TG_NAME

Tipo de dado name; variável contendo o nome do gatilho disparado.

TG_WHEN

Tipo de dado text; uma cadeia de caracteres contendo BEFORE ou AFTER, dependendo da definição do gatilho.

TG_LEVEL

Tipo de dado text; uma cadeia de caracteres contendo ROW ou STATEMENT, dependendo da definição do gatilho.

TG_OP

Tipo de dado text; uma cadeia de caracteres contendo INSERT, UPDATE, ou DELETE, informando para qual operação o gatilho foi disparado.

TG_RELID

Tipo de dado oid; o ID de objeto da tabela que causou o disparo do gatilho.

TG_RELNAME

Tipo de dado name; o nome da tabela que causou o disparo do gatilho.

TG_NARGS

Tipo de dado integer; o número de argumentos fornecidos ao procedimento de gatilho na instrução CREATE

TG_ARGV[]

Tipo de dado matriz de text; os argumentos da instrução CREATE TRIGGER. O contador do índice começa por 0.

Índices inválidos (menor que 0 ou maior ou igual a tg_nargs) resultam em um valor nulo.

Exemplo:

O gatilho deste exemplo garante que:

- quando é inserida ou atualizada uma linha na tabela, fica sempre registrado nesta linha o usuário que efetuou a inserção ou a atualização

- quando isto ocorreu.

- além disso, o gatilho verifica se é fornecido o nome do empregado

- e se o valor do salário é um número positivo.

```
CREATE TABLE emp (
        nome_emp text,
        salario integer,
        ultima_data timestamp,
        ultimo_usuario text
);

CREATE FUNCTION emp_gatilho() RETURNS trigger AS $emp_gatilho$
        BEGIN
                -- Verificar se foi fornecido o nome e o salário do empregado
                IF NEW.nome_emp IS NULL THEN
                        RAISE EXCEPTION 'O nome do empregado não pode ser nulo';
                END IF;
                IF NEW.salario IS NULL THEN
                        RAISE EXCEPTION '% não pode ter um salário nulo',
NEW.nome_emp;
                END IF;
                -- Quem paga para trabalhar?
                IF NEW.salario < 0 THEN
                        RAISE EXCEPTION '% não pode ter um salário negativo',
NEW.nome_emp;
                END IF;
                -- Registrar quem alterou a folha de pagamento e quando
                NEW.ultima_data := 'now';
                NEW.ultimo_usuario := current_user;
                RETURN NEW;
        END;

$emp_gatilho$ LANGUAGE plpgsql;

CREATE TRIGGER emp_gatilho BEFORE INSERT OR UPDATE ON emp
        FOR EACH ROW EXECUTE PROCEDURE emp_gatilho();

INSERT INTO emp (nome_emp, salario) VALUES ('João',1000);
INSERT INTO emp (nome_emp, salario) VALUES ('José',1500);
INSERT INTO emp (nome_emp, salario) VALUES ('Maria',2500);

SELECT * FROM emp;
```

Gatilho para registrar inserções e atualizações

```
CREATE TABLE emp (
        nome_emp text,
        salario integer,
        usu_cria text, -- Usuário que criou a linha
        data_cria timestamp, -- Data da criação da linha
        usu_atu text, -- Usuário que fez a atualização
        data_atu timestamp -- Data da atualização
);

CREATE FUNCTION emp_gatilho() RETURNS trigger AS $emp_gatilho$
        BEGIN
                -- Verificar se foi fornecido o nome do empregado
                IF NEW.nome_emp IS NULL THEN
                        RAISE EXCEPTION 'O nome do empregado não pode ser nulo';
                END IF;
                IF NEW.salario IS NULL THEN
                        RAISE EXCEPTION '% não pode ter um salário nulo',
NEW.nome_emp;
                END IF;
                -- Quem paga para trabalhar?
                IF NEW.salario < 0 THEN
                        RAISE EXCEPTION '% não pode ter um salário negativo',
NEW.nome_emp;
                END IF;
                -- Registrar quem criou a linha e quando
                IF (TG_OP = 'INSERT') THEN
                        NEW.data_cria := current_timestamp;
                        NEW.usu_cria := current_user;
                -- Registrar quem alterou a linha e quando
                ELSIF (TG_OP = 'UPDATE') THEN
                        NEW.data_atu := current_timestamp;
                        NEW.usu_atu := current_user;
                END IF;
                RETURN NEW;
END;
$emp_gatilho$ LANGUAGE plpgsql;

CREATE TRIGGER emp_gatilho BEFORE INSERT OR UPDATE ON emp
        FOR EACH ROW EXECUTE PROCEDURE emp_gatilho();

INSERT INTO emp (nome_emp, salario) VALUES ('João',1000);
INSERT INTO emp (nome_emp, salario) VALUES ('José',1500);
INSERT INTO emp (nome_emp, salario) VALUES ('Maria',250);
UPDATE emp SET salario = 2500 WHERE nome_emp = 'Maria';

SELECT * FROM emp;
```

Gatilho para auditoria

Todas as operações na tabela emp serão registradas na tabela emp_audit

```
CREATE TABLE emp (
        nome_emp text NOT NULL,
        salario integer
);

CREATE TABLE emp_audit(
        operacao char(1) NOT NULL,
        usuario text NOT NULL,
        data timestamp NOT NULL,
        nome_emp text NOT NULL,
        salario integer
);

CREATE OR REPLACE FUNCTION processa_emp_audit() RETURNS TRIGGER AS $emp_audit$
        BEGIN
                --
                -- Cria uma linha na tabela emp_audit para refletir a operação
                -- realizada na tabela emp. Utiliza a variável especial TG_OP
                -- para descobrir a operação sendo realizada.
                --
                IF (TG_OP = 'DELETE') THEN
                        INSERT INTO emp_audit SELECT 'E', user, now(), OLD.*;
                        RETURN OLD;
                ELSIF (TG_OP = 'UPDATE') THEN
                        INSERT INTO emp_audit SELECT 'A', user, now(), NEW.*;
                        RETURN NEW;
                ELSIF (TG_OP = 'INSERT') THEN
                        INSERT INTO emp_audit SELECT 'I', user, now(), NEW.*;
                        RETURN NEW;
                END IF;
                RETURN NULL; -- o resultado é ignorado uma vez que este é um
gatilho AFTER
        END;
$emp_audit$ language plpgsql;

CREATE TRIGGER emp_audit
AFTER INSERT OR UPDATE OR DELETE ON emp
        FOR EACH ROW EXECUTE PROCEDURE processa_emp_audit();

INSERT INTO emp (nome_emp, salario) VALUES ('João',1000);
INSERT INTO emp (nome_emp, salario) VALUES ('José',1500);
INSERT INTO emp (nome_emp, salario) VALUES ('Maria',250);
UPDATE emp SET salario = 2500 WHERE nome_emp = 'Maria';
DELETE FROM emp WHERE nome_emp = 'João';

SELECT * FROM emp;

SELECT * FROM emp_audit;
```

Autoditoria ao nível de campos

```
CREATE TABLE emp (
        id serial PRIMARY KEY,
        nome_emp text NOT NULL,
        salario integer
);

CREATE TABLE emp_audit(
        usuario text NOT NULL,
        data timestamp NOT NULL,
        id integer NOT NULL,
        coluna text NOT NULL,
        valor_antigo text NOT NULL,
        valor_novo text NOT NULL
);

CREATE OR REPLACE FUNCTION processa_emp_audit() RETURNS TRIGGER AS $emp_audit$
BEGIN
                --
                -- Não permitir atualizar a chave primária
                --
                IF (NEW.id <> OLD.id) THEN
                        RAISE EXCEPTION 'Não é permitido atualizar o campo ID';
                END IF;
                --
                -- Inserir linhas na tabela emp_audit para refletir as
alterações
                -- realizada na tabela emp.
                --
                IF (NEW.nome_emp <> OLD.nome_emp) THEN
                        INSERT INTO emp_audit SELECT current_user,
current_timestamp,
                        NEW.id, 'nome_emp', OLD.nome_emp, NEW.nome_emp;
                END IF;
                IF (NEW.salario <> OLD.salario) THEN
                        INSERT INTO emp_audit SELECT current_user,
current_timestamp,
                        NEW.id, 'salario', OLD.salario, NEW.salario;
                END IF;
                RETURN NULL; -- o resultado é ignorado uma vez que este é um
gatilho AFTER
        END;
$emp_audit$ language plpgsql;

CREATE TRIGGER emp_audit
AFTER UPDATE ON emp
        FOR EACH ROW EXECUTE PROCEDURE processa_emp_audit();

INSERT INTO emp (nome_emp, salario) VALUES ('João',1000);
INSERT INTO emp (nome_emp, salario) VALUES ('José',1500);
INSERT INTO emp (nome_emp, salario) VALUES ('Maria',2500);
UPDATE emp SET salario = 2500 WHERE id = 2;
UPDATE emp SET nome_emp = 'Maria Cecília' WHERE id = 3;
UPDATE emp SET id=100 WHERE id=1;
ERRO: Não é permitido atualizar o campo ID

SELECT * FROM emp;
```

```
SELECT * FROM emp_audit;
```

Gatilho para manter uma tabela sumário

O esquema que está detalhado a seguir é parcialmente baseado no exemplo Grocery Store do livro The Data Warehouse Toolkit de Ralph Kimball.

```
--
-- Main tables - time dimension and sales fact.
--

CREATE TABLE time_dimension (
        time_key integer NOT NULL,
        day_of_week integer NOT NULL,
        day_of_month integer NOT NULL,
        month integer NOT NULL,
        quarter integer NOT NULL,
        year integer NOT NULL
);

CREATE UNIQUE INDEX time_dimension_key ON time_dimension(time_key);

CREATE TABLE sales_fact (
        time_key integer NOT NULL,
        product_key integer NOT NULL,
        store_key integer NOT NULL,
        amount_sold numeric(12,2) NOT NULL,
        units_sold integer NOT NULL,
        amount_cost numeric(12,2) NOT NULL
);

CREATE INDEX sales_fact_time ON sales_fact(time_key);

--
-- Summary table - sales by time.
--

CREATE TABLE sales_summary_bytime (
        time_key integer NOT NULL,
        amount_sold numeric(15,2) NOT NULL,
        units_sold numeric(12) NOT NULL,
        amount_cost numeric(15,2) NOT NULL
);

CREATE UNIQUE INDEX sales_summary_bytime_key ON sales_summary_bytime(time_key);

--
-- Function and trigger to amend summarized column(s) on UPDATE, INSERT, DELETE.
--

CREATE OR REPLACE FUNCTION maint_sales_summary_bytime() RETURNS TRIGGER AS
$maint_sales_summary_bytime$
        DECLARE
                delta_time_key integer;
                delta_amount_sold numeric(15,2);
                delta_units_sold numeric(12);
                delta_amount_cost numeric(15,2);
```

```
        BEGIN
                -- Work out the increment/decrement amount(s).
                IF (TG_OP = 'DELETE') THEN
                        delta_time_key = OLD.time_key;
                        delta_amount_sold = -1 * OLD.amount_sold;
                        delta_units_sold = -1 * OLD.units_sold;
                        delta_amount_cost = -1 * OLD.amount_cost;
                ELSIF (TG_OP = 'UPDATE') THEN
                        -- forbid updates that change the time_key -
                        -- (probably not too onerous, as DELETE + INSERT is how
most
                        -- changes will be made).
                        IF ( OLD.time_key != NEW.time_key) THEN
                                RAISE EXCEPTION 'Update of time_key : % -> % not
allowed', OLD.time_key,
                                NEW.time_key;
                        END IF;
                        delta_time_key = OLD.time_key;
                        delta_amount_sold = NEW.amount_sold - OLD.amount_sold;
                        delta_units_sold = NEW.units_sold - OLD.units_sold;
                        delta_amount_cost = NEW.amount_cost - OLD.amount_cost;
                ELSIF (TG_OP = 'INSERT') THEN
                        delta_time_key = NEW.time_key;
                        delta_amount_sold = NEW.amount_sold;
                        delta_units_sold = NEW.units_sold;
                        delta_amount_cost = NEW.amount_cost;
                END IF;
                -- Update the summary row with the new values.
                UPDATE sales_summary_bytime
                        SET amount_sold = amount_sold + delta_amount_sold,
                                units_sold = units_sold + delta_units_sold,
                                amount_cost = amount_cost + delta_amount_cost
                WHERE time_key = delta_time_key;

                -- There might have been no row with this time_key (e.g new
data!).
                IF (NOT FOUND) THEN
                        BEGIN
                                INSERT INTO sales_summary_bytime (
                                        time_key,
                                        amount_sold,
                                        units_sold,
                                        amount_cost)
                        VALUES (
                                delta_time_key,
                                delta_amount_sold,
                                delta_units_sold,
                                delta_amount_cost
                                );
                        EXCEPTION
                                --
                                -- Catch race condition when two transactions
are adding data
                                -- for a new time_key.
                                --
                                WHEN UNIQUE_VIOLATION THEN
                                        UPDATE sales_summary_bytime
                                                SET amount_sold = amount_sold +
delta_amount_sold,
```

```
                                                        units_sold = units_sold
+ delta_units_sold,
                                                        amount_cost =
amount_cost + delta_amount_cost
                                                  WHERE time_key = delta_time_key;
                        END;
                END IF;
                RETURN NULL;
        END;
$maint_sales_summary_bytime$ LANGUAGE plpgsql;

CREATE TRIGGER maint_sales_summary_bytime
AFTER INSERT OR UPDATE OR DELETE ON sales_fact
        FOR EACH ROW EXECUTE PROCEDURE maint_sales_summary_bytime();
```

Este é um resumo da documentação oficial em inglês e em português do Brasil, que devem ser consultados para informações mais detalhadas em:

- http://www.postgresql.org/docs/8.2/interactive/index.html

- http://pgdocptbr.sourceforge.net/pg80/index.html

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Administra%C3%A7%C3%A3o_dos_SGBDs/PostgreSQL"
Page categories: PHP

Aplicativos em PHP/Administração dos SGBDs/SQLite

De Wikibooks

< Aplicativos em PHP | Administração dos SGBDs

Conteúdo

- 1 SQLite na Wikipedia
- 2 Características atuais
- 3 Criando Banco
- 4 Efetuando Consultas
- 5 Usando SQLite com PHP
- 6 Exemplo de aplicativo simples em PHP acessando SQLite

SQLite na Wikipedia

http://pt.wikipedia.org/wiki/Sqlite

SQLite é uma biblioteca C que implementa um banco de dados SQL embutido. Programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo RDBMS separado.

SQLite não é uma biblioteca de cliente usada para conectar com um grande servidor de banco de dados. SQLite é o servidor. A biblioteca SQLite lê e escreve diretamente para e do arquivo do banco de dados no disco.

O uso do SQLite é recomendado onde a simplicidade da administração, implementação e manutenção são mais importantes que incontáveis recursos que SGBDs mais voltados para aplicações complexas possivelmente implementam. Entretando situações onde a simplicidade é a melhor escolha são muito mais freqüentes do que pode-se imaginar.

Exemplos de uso do SQLite são, não restrito a, sites (com menos de cem mil requisições por dia), dispositivos e sistemas embarcados, aplicações desktop, ferramentas estatísticas e de análise, aprendizado de banco de dados, implementação de novas extensões à SQL. Não se recomenda o uso do SQLite para sites com muitos acessos, grande quantidades de dados (talvez maior que algumas duzias de gigabytes), sistemas com grande concorrência, aplicações cliente/servidor.

Programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo RDBMS separado.

A biblioteca SQLite lê e escreve diretamente para e do arquivo de banco de dados no disco.

Características atuais

- Transações são atômicas, consistentes, isoladas e duráveis (ACID) mesmo que o sistema trave ou a energia falhe.

- Configuração-zero - nenhuma instalação ou administração necessária.

- Implementação da maior parte do SQL92.

- Um banco de dados completo é armazenado em apenas um arquivo de sistema.

- Arquivos de banco de dados podem ser livremente compartilhados entre máquinas com diferentes ordens de byte.

- Suporta bases de dados de até 2 terabytes de tamanho.

- Tamanho de strings e BLOBs limitados apenas pela memória disponível.

- Mais rápido que populares bancos de dados cliente/servidor para a maioria das operações comuns.

- API simples e fácil de usar.

- TCL bindings inclusas. Bindings para a maioria das linguagens disponíveis separadamente.

- Código fonte bem comentado, com mais de 95% coberto por testes.

- Auto-contido: sem dependências externas.

- Fontes estão em domínio público. Use para qualquer propósito.

A distribuição SQLite vem com um programa de linha de comando (sqlite (http://sqlitebrasil.codigolivre.org.br/?pagina=doc/sqlite)) que pode ser usado para administrar um banco de dados SQLite e que serve como exemplo de como usar a biblioteca SQLite.

Além do programa em linha de comando, você pode utilizar alguns dos programas de terceiros com interface gráfica, como o SQLiteManager (http://sqlitemanager.sourceforge.net/) (web, no estilo PHPMyAdmin) ou o SQLiteBrowser (http://sqlitebrowser.sourceforge.net/) (QT).

O SQLite está embutido no PHP (http://br.php.net/sqlite) 5 e disponível como extensão no PHP 4. Assim, qualquer aplicação PHP pode utilizar um banco de dados sem necessitar de um SGDB.

Há um driver (http://dba.openoffice.org/drivers/sqlite/index.html) (alpha) disponível para conectar bancos de dados SQLite com o OpenOffice.org. Tornando possível a criação de relatórios, formulários, etc.

Suas características o tornam ideal para desenvolver programas standalone, pequenos e médios sites, etc. Veja quando usar (e não usar) (http://localhost/sqlitewww/?pagina=doc/quando) o SQLite.

Criando Banco

sqlite clientes (Com este comando ele cria o banco clientes e já acessa a console deste banco)

Criando Tabela (A sintaxe não tem diferença dos grandes SGBDs, é puro SQL)

```
CREATE TABLE cliente (
        cpf     VARCHAR(11) PRIMARY KEY,
        nome VARCHAR(45),
        fone VARCHAR(10)
);
```

Efetuando Consultas

Inserindo Registros (Também puro SQL)

INSERT INTO cliente (cpf, nome, fone) VALUES ('11111111111', 'João Abreu', '34543456');

SELECT * FROM cliente;

Outros comandos do sqlite. Estando na console apenas digite ".help":

```
sqlite> .help
.databases             List names and files of attached databases
.dump ?TABLE? ...      Dump the database in a text format
.echo ON|OFF           Turn command echo on or off
.exit                  Exit this program
.explain ON|OFF        Turn output mode suitable for EXPLAIN on or off.
.header(s) ON|OFF      Turn display of headers on or off
```

```
.help                   Show this message
.indices TABLE          Show names of all indices on TABLE
.mode MODE              Set mode to one of "line(s)", "column(s)",
                        "insert", "list", or "html"
.mode insert TABLE      Generate SQL insert statements for TABLE
.nullvalue STRING       Print STRING instead of nothing for NULL data
.output FILENAME        Send output to FILENAME
.output stdout          Send output to the screen
.prompt MAIN CONTINUE   Replace the standard prompts
.quit                   Exit this program
.read FILENAME          Execute SQL in FILENAME
.schema ?TABLE?         Show the CREATE statements
.separator STRING       Change separator string for "list" mode
.show                   Show the current values for various settings
.tables ?PATTERN?       List names of tables matching a pattern
.timeout MS             Try opening locked tables for MS milliseconds
.width NUM NUM ...      Set column widths for "column" mode
```

Escrevendo o Resultado em um Arquivo:

```
sqlite> .mode list
sqlite> .separator |
sqlite> .output test_file_1.txt
sqlite> select * from tbl1;
sqlite> .exit
$ cat test_file_1.txt
hello|10
goodbye|20
```

Usando SQLite com PHP

```
<?php

$db=sqlite_open("/home/ribafs/bancos/clientes.db");

$registros = sqlite_query($db, "SELECT * FROM cliente");

while ($i = sqlite_fetch_array($registros)) {
        print $i[0] . "__" . $i[1]. "__" . $i[2]. "<br>";
}

?>
```

Exemplo de aplicativo simples em PHP acessando SQLite

http://www.ribafs.net/down/appsexemplo/sqlite/iniciante_sqlitephp.zip

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Administra%C3%A7%C3%A3o_dos_SGBDs/SQLite"
Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos

De Wikibooks

< Aplicativos em PHP

Geradores de Aplicativos

"Um livro é um mudo que fala, um surdo que responde, um cego que guia, um morto que vive." (Padre António Vieira)

1. phpCodeGenie (com MySQL)
2. SQLMaestro (com MySQL, Oracle, MS SQL Server, PostgreSQL, SQLite, Firebird e MaxDB)
3. phpMyEdit (com MySQL)
4. DadaBik (com vários)
5. PHPLibDev (com PostgreSQL)
6. Web Form Generator (com MySQL)
7. PHP Code Generator (com MySQL, PostgreSQL, Access e logo outros)

Segundo a wikipédia em http://pt.wikipedia.org/wiki/geradores

Dentro do diversificado leque de categorias de ferramentas que prestam apoio às atividades da Engenharia de Software (CASE), uma específica vem ganhando cada vez mais destaque e, sobre ela, tem-se aplicado muito investimento nos últimos tempos: as Ferramentas de Geração de Código, ou simplesmente Geradores de Código. Destes surgiu o Gerador de Aplicativos.

Dessa forma, Gerador de Código é aquela ferramenta que possui a capacidade de gerar código a partir de um determinado modelo de software. Inclusive, de acordo com alguns pontos de vista e a partir das características específicas do tipo de Gerador de Código, ele passa a ser conversor de códigos de linguagens distintas. Isso acontece, por exemplo, com o compilador, que transforma um código escrito através de uma linguagem de programação para código de máquina ou código objeto.

Tipos mais comuns

Alguns tipos de Gerador de Código valem a pena serem mencionados. Eles possuem características específicas e uma maneira de trabalhar própria que está diretamente ligada aos seus objetivos. O mais básico tipo de gerador de código é o já mencionado compilador, como pode ser visto no artigo Code generation.

Outros dois tipos de gerador de código que estão sendo cada vez mais utilizados em ambiente de produção de software são os geradores de código de IDE's visuais e os geradores de código baseados em templates.

Geradores de código de IDE's visuais

Algumas ferramentas de edição integrada (IDE's), principalmente as chamadas ferramentas de programação visual, precisaram adotar um tipo de gerador de código capaz de traduzir, em linhas de código, tudo aquilo que é "desenhado" pelo usuário desenvolvedor. Dessa maneira, por trás de ações como arrastar componentes para uma janela de uma aplicação que está sendo construída e configurar visualmente seu comportamento e forma de apresentação, vai existir um mecanismo responsável por transformar em instruções que, quando executadas, reproduzam exatamente aquilo que foi definido pelo usuário desenvolvedor, através de linhas de código da linguagem de programação correspondente à ferramenta IDE. São os também chamados mecanismos WYSIWYG. Tal mecanismo pode ser encontrado em ferramentas como o Borland Delphi, o Oracle JDeveloper ou o Microsoft Visual Studio, quando do desenvolvimento de aplicações tanto de ambiente desktop, como Web.

Geradores de código baseados em templates

Ultimamente, os geradores de código baseados em templates começaram a figurar o elenco das ferramentas CASE presentes em ambiente de produção de software como sendo uma opção para a geração automática e massiva de funcionalidades de aplicações, seja de maneira completa ou parcial.

Ferramentas tais como Velocity (do Apache Jakarta Project), Transformica (do Hammurapi Group) ou CodeFSW (da Unitech Tecnologia de Informação), além de diversas outras que vêm surgindo neste cenário, a depender de como sejam utilizadas, podem viabilizar a geração de código para as diversas camadas de uma aplicação, a partir de templates e parâmetros de entrada. Dessa maneira, tendo a necessidade de possuir diversas funcionalidades semelhantes em suas estrutura e que englobem códigos das camadas da aplicação adotadas em sua arquitetura, uma aplicação pode ser desenvolvida, em grande parte, com a ajuda de tais ferramentas.

Principais vantagens

Certamente que as vantagens adquiridas ao se adotar Geradores de Código em um ambiente de produção de software só aparecem quando da boa utilização dessas ferramentas. A partir do uso adequado dos geradores de código, sejam quais tipos forem, a equipe de produção de software perceberá um ganho significativo em sua produtividade, sendo capaz de realizar a construção de funcionalidades das aplicações de maneira mais eficiente e segura, garantindo também a diminuição dos custos de projeto.

Uma outra vantagem que é vista de maneira bastante clara se refere à padronização percebida no código gerado, visto que pode se basear, a depender do tipo, tanto em um template utilizado, como na filosofia de trabalho da ferramenta IDE. E está definitivamente consolidada a idéia de que é um ponto positivo com relação à qualidade do software, ter uma aplicação em cujo código-fonte se percebe um estilo único de programação.

Por fim, pode-se considerar também como uma vantagem do uso de geradores de código, a depender novamente de como são utilizados, a construção de aplicações isentas de grande parte dos artifícios que facilitam a programação propriamente, mas, por outro lado, afetam diretamente o quesito performance. É sabido, por exemplo, que o uso de Reflexão, artifício que é suportado em linguagens como Java e C# e que permite a introspecção e manipulação de classes, interfaces e objetos no geral, afeta negativamente a performance de funcionalidades de aplicações. O uso adequado de geradores de código no processo de desenvolvimento de software pode fazer desnecessário o uso de tal artifício, melhorando o desempenho do produto final. Muitas outras vantagens podem ser percebidas com o uso de geradores de código no processo de construção de

software. Mas também vale alertar que tais ferramentas, em vez de contribuir com vantagens, podem atrapalhar o andamento do projeto quando é feito uso inadequado ou desnecessário das mesmas.

Geradores de código em PHP

Como para diversas outras linguagens, também existem bons geradores de código em PHP, existindo ferramentas disponíveis em software livre e software proprietário. Daremos maior ênfase aquelas da categoria software livre, que estão mais de acordo com a filosofia deste projeto e também porque apresentam resultados semelhantes e as vezes até melhores do que as outras.

A grande maioria dos geradores de código existentes em PHP, trabalha de forma similar quando da construção das aplicações. Necessitam que um banco de dados já construído e de uma conexão com esse banco de dados e a partir das informações aí existentes, geram o código para a aplicação. Poderíamos também classificar os geradores de código em PHP em dois diferentes modelos: ativo e passivo.

No modelo ativo, o código se mantém enquanto o gerador estiver ativo, ou seja, para que o aplicativo funcione, é necessária a presença do gerador ou pelo menos de parte integrante dele (um bom exemplo nesse caso é o phpMyEdit). Já no modelo passivo, o gerador cria o código inicial e sua manutenção e funcionalidade passa a depender completamente do desenvolvedor, o PHP-MySQL Wizard pode ser enquadrado nesse modelo. Existem geradores que trabalham somente com OO, linguagem procedural ou um misto.

Vantagens adicionais:

Conforme artigo no code generation, existem vantagens adicionais em se utilizar geradores de código:

1.Padrão de qualidade: o código feito a mão tem sua qualidade variando durante o ciclo de vida de um projeto. Pode começar no alto e decair ou vice e versa. O código gerado aumenta sua qualidade com o tempo pois os erros encontrados podem ser uniformemente reparados na base do código. Nesse quesito, os geradores que têm projetos ativos e que constantemente lançam novas versões ganham pontos. A familiaridade com o gerador por parte do programador também ajuda bastante, visto que dependendo dos seus conhecimentos, ele mesmo pode fazer reparos ou customizações na ferramenta (quando open source) e não somente no código gerado.

2.Consistência: as aplicações geradas são consistentes na estrutura de classes, nomeação de variáveis etc.

3.Produtividade: os geradores constroem o código numa fração muita curta de tempo, economizando horas de trabalho (muitas vezes repetitivo). Isso libera o desenvolvedor para outras tarefas que exijam soluções mais criativas.

4.Abstração: Os geradores frequentemente fornecem uma camada de abstração entre o projeto e a base do código, facilitando o manuseio das regras de negócio.

É interessante notar que entre os desenvolvedores médios e avançados em PHP e creio que também nas outras linguagens, existe uma certa resistência ou menosprezo pelo uso de geradores. É importante estarmos abertos para utilizarmos quaisquer ferramentas que melhorem nossa produtividade, nos poupem trabalho repetitivo e pouco criativo, deixem nosso tempo livre para outras atividades e consequentemente melhorem nossa qualidade de vida. Assim, antes de iniciar seu projeto você deve se perguntar: que ferramentas usarei? Esse ou aquele gerador me será útil em alguma fase? Ele funciona corretamente? Cumpre sua finalidade? Se a resposta é sim, vá em frente

e esqueça o preconceito. Com certeza existem projetos para os quais compensa até construir um gerador. Ressalte-seainda que os geradores são somente ferramentas e a qualidade do código gerado vai depender diretamente da sua habilidade em manuseá-las, aplicar ajustes necessários, etc. Como em todo projeto de criação de software, aqueles criados com geradores de código apresentam falhas. Baterias exaustivas de testes, conhecimento profundo da ferramenta e bom planejamento ajudam a corrigí-las. Uma dica valiosa é dar preferência a geradores com projetos ativos, boa documentação, fóruns e/ou listas de discussão. Hoje, alguns dos melhores frameworks em PHP tais como o Cake, Symfony e CodeIgniter possuem ferramentas para gerar código entre suas funcionalidades.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos"
Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos/phpCodeGenie (com MySQL)

De Wikibooks

< Aplicativos em PHP | Geradores de Aplicativos

phpCodeGenie (PCG)

Um excelente gerador de código em PHP é o phpCodeGenie (PCG). Ele é um gerador do tipo passivo e permite níveis bastante altos de customização. Conforme a apresentação do PCG "Melhor que gastar muito tempo criando código comum, é poder gastá-lo na lógica de negócios de nossas aplicações e deixar PCG fazer o código aborrecido, pois o phpCodeGenie pode gerar o código simples para os novatos onde tudo é feito no mesmo script PHP ou o código orientado a objeto da estrutura PHP, que segue o phpCodeGenie da estrutura de PCG que requer projetar suas tabelas e então o gênio pode gerar o código para a leitura e a escrita à base de dados, os formulários do HTML para incorporar ou editar dados à base de dados, scripts para listar dados, scripts para excluir dados, formulários de busca, scripts de busca entre outros. O phpCodeGenie foi projetado com o usuário iniciante em mente como também com o programador experiente em PHP que quer gerar uma aplicação complexa."

Infelizmente hoje o PCG é um projeto inativo, mas foi desenvolvido o bastante para ainda ser bastante útil. Os requisitos para instalação são básicos: servidor web, PHP e MySQL instalados. A última versão que temos conhecimento é a 3.0.2. Após, o responsável pelo projeto, Nilesh Dosooye, abandonou o mesmo.

Principais características:

- É um gerador do tipo passivo. Basta gerar o código e fazer as modificações necessárias. Ele pode ser livremente transferido para outros diretórios, apenas ajustando-se o path nos scripts common/footer.php e common/header.php.

- É opensource e licenciado sob GNU – GPL . Compatível com Linux e Windows .

- Não possui muita documentação.

- Só funciona com MySQL, embora aparentemente o autor tivesse a pretensão de dar suporte a outros SGBDs.

- O código pode ser gerado individualmente para uma tabela de cada vez ou para um banco inteiro.

- Quando da geração do CRUD simples, o PCG gera uma página (ou mais) para cada função. No caso do CRUD, o código gerado requer alguns ajustes, principalmente de natureza estética.

- O download e instalação apresentam alguma dificuldade. O acesso ao gerador se dá via browser.

- Não existe comunidade ativa para o projeto (pelo menos de nosso conhecimento).

- O Projeto está inativo, o que significa que você mesmo terá que se virar com bugs, etc.

- É bastante customizável, exigindo no entanto o mínimo de conhecimento de html e CSS, bons conhecimentos de linguagem PHP procedural ou OO.

Possui tradução para o português, por Ribamar FS.

Corrigindo um pequeno Bug

Edite o script common/footer.php e elimine o fechamento de tag:

Que é a primeira linha do script. Sempre terá que remover esta linha dos aplicativos gerados ou procurar a ocorrência no gerador e corrigir.

Instalação e utilização

Download Souce Forge:[1] No site do ribafs [2], você encontra um bom tutorial de instalação e o arquivo com a tradução para o português.

Após instalar e configurar, você deve ter a seguinte estrutura de pastas:

Os aplicativos gerados na opção programadores iniciantes, por exemplo, ficam na sub-pasta \web\generatedCode\simple\nome_do_aplicativo. Essa opção default, pode ser modificada.

Se não houve problemas, quando acessar o endereço do PCG via browser, você terá uma tela assim:

Preenchendo os dados para conexão, você é remetido a uma nova tela onde aparecem os bancos existentes no SGBD selecionado. Daí, é escolher a opção desejada e com alguns cliques, seu código está pronto. Você tem opções de gerar por exemplo todo o CRUD ou somente parte dele. Fazer isso utilizando linguagem procedural ou Orientação a Objeto etc.

Para gerar um aplicativo completo utiliza o PHP procedural e para gerar alguns utilitários utiliza o PHPOO.

Abaixo, tela mostrando as opções do PCG após a geração de um aplicativo.

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos/phpCodeGenie_%28com_MySQL%29"
Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos/SQLMaestro (com MySQL, Oracle, MS SQL Server, PostgreSQL, SQLite, Firebird e MaxDB)

De Wikibooks

< Aplicativos em PHP | Geradores de Aplicativos

7.2 - SQLMaestro (MySQL, Oracle, MS SQL Server, PostgreSQL, SQLite, Firebird e MaxDB)

Geradores de aplicativos em PHP com os principais SGBDs do mercado.

Site oficial - http://www.sqlmaestro.com/download/

Os geradores para todos os SGBDs no site são free.

Basta fazer o download, instalar e usar.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos/SQLMaestro_%28com_MySQL%2C_Oracle%2C_MS_SQL_Server%2C_PostgreSQL%2C_SQLite%2C_Firebird_e_MaxDB%29"
Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos/phpMyEdit

De Wikibooks

< Aplicativos em PHP | Geradores de Aplicativos

7.3 - phpMyEdit

Em se tratando de geradores de código CRUD (acrônimo para Create, Retrieve, Update e Delete) para PHP, uma das melhores opções é o phpMyEdit. Como os desenvolvedores do projeto informam na página de abertura, o phpMyEdit é provavelmente a maneira mais rápida de gerar formulários para interagir com uma base de dados MySQL.

O phpMyEdit gera o código para edição de tabelas no MySQL com uma enorme gama de funções de manipulação (inserção, alterações, visualização, cópia, remoção de registros inseridos). Para isso você necessita apenas criar um script de chamada das funções (o phpMyEdit traz uma aplicação já pronta para desenvolver esse arquivo). Você pode ainda executar buscas com filtro, lookups etc. O gerador é open source e liberado sob licença GNU-GPL ou licença comercial. Ainda conforme os desenvolvedores, ele é utilizado em diversos sistemas ativos (há referências no site). Os requisitos são os básicos: servidor web (os desenvolvedores recomendam o Apache), PHP e MySQL instalados.

Principais características:

É um gerador do tipo ativo. Para que seu aplicativo funcione, no mínimo o arquivo phpMyEdit.class.php e os arquivos de linguagem devem estar presentes no seu diretório.

É opensource e licenciado sob GNU – GPL ou licença comercial.

Compatível com Linux, Windows e outros.

Possui boa documentação e tutoriais, disponiveis no site.

Só funciona com MySQL. O código é gerado individualmente para uma tabela de cada vez.

O download e instalação são bastante fáceis. O acesso ao gerador se dá via browser

Existe uma comunidade ativa com fóruns e listas de discussão que ajudam no desenvolvimento, dúvidas e outros.

O Projeto está ativo, o que significa melhorias, correções de eventuais bugs etc.

É bastante customizável, exigindo no entanto o mínimo de conhecimento de OO, html e java script.

Possui um arquivo de tradução para o português, exibindo botões, chamadas e mensagens em nossa língua. Esse arquivo pode ser modificado e merece algumas correções, porém mais de 90% do trabalho já está pronto.

Instalação e utilização

Faça o Download em

Descompacte o arquivo em seu diretório web (Se no Linux ou similar dê permissão de escrita no diretório do phpMyEdit).

Após isso, você deve ter uma estrutura de pastas como essa:

Agora, acesse pelo browser o arquivo phpMyEditSetup.php

Se tudo correu bem, você vai ter uma tela como esta:

Preenchendo os dados relativos ao SGGB, o phpMyEdit realiza a conexão e já deve mostrar os dados da tabela. Veja que é recomendado ter um campo de identificação para ser utilizado como identificador nas operações (editar, adicionar etc). Recomenda-se também que este campo precisa ser numérico e único, a chave primária da tabela, sendo uma boa idéia nomeá-lo como "id". Dependendo da sua tabela, você pode criar um campo autoincrement. No exemplo, estou usando a tabela dados_atendimento.

Em seguida, o phpMyEdit vai lhe apresentar a tela com opções adicionais. Você pode gerar um cabeçalho (título) para a página, o html de um cabeçalho e rodapé e uma folha de estilo(CSS básico). Tudo isso é gerado em um único arquivo. Por default, vem marcada a penas a opção da folha de estilo. Caso ache as opções do CSS limitadas,você pode editá-las acrescentando ou modificando, bem como utilizar outro arquivo de CSS, bastando vinculá-lo à página.

Bom, depois dessa série simples de cliques, você já tem seu aplicativo pronto para usar. Veja que o

phpMyEdit salva o arquivo PHP do aplicativo em sua pasta base e com o nome da tabela. Ele também apresenta no browser o código gerado. Você pode copiá-lo e criar um novo arquivo em outtra pasta. Como já dissemos, nesse caso, você deve mover para essa mesma pasta pelo menos o arquivo phpMyEdit.class.php e a pasta lang. Se na configuração quiser usar imagens em vez dos tradicionais botões Javascript, mova também a pasta images. O phpMyEdit possui uma série de estensões como cal pop-up (calendário), messages etc. Caso queira usá-las, mova também a pasta extensions.

Essa é a interface básica do arquivo gerado. Note que os nomes dos campos são idênticos aos existentes no banco de dados. Com relação à linguagem, o phpMyEdit já roda o aplicativo com a configuração de linguagem padrão do browser (testei com o Mozilla Firefox 2 e o Internet Explorer 6 e 7 (ou seja se seu seu navegador está em português, é nessa linguagem que você verá seu aplicativo). O botão em destaque abre a opção de busca, onde você pode utilizar mais de um campo como filtro para as pesquisas.

Editando o arquivo criado, você pode alterar as opções para gerenciamento do CRUD, deixando por exemplo somente a visualização. Também pode alterar a posição dos botões colocando-os na parte superior ou inferior do aplicativo ou até em ambas, estipular a quantidade de registros a ser exibida em cada página, definir se utilizará imagens em vez de botões tradicionais ou em conjunto com eles, modificar o nome dos campos da tabela para exibição etc. Como já foi dito, é possível modifcar o CSS ou criar um novo. Também podemos configurar mensagens em javascript para, por exemplo, quando um campo da tabela for not null, avisar ao usuário que o mesmo deve ser preenchido. Todas essas opções são facilmente customizáveis. O arquivo gerado vem todo comentado, o que facilita muito. Como estamos usando PHP, obviamente também é podemos modificar a forma de exibição de datas, já que no padrão MySQL, elas vêm na forma ano- mês-dia. Enfim, o phpMyEdit é um ótimo gerador de código e editor de tabelas, repleto de possibilidades. No código abaixo, simplesmente suprimindo ou acrescentando letras, já modificamos as opções de exibição: no primeiro trecho em destaque, estamos permitindo somente adicionar, ver e modificar os registros. No segundo trecho, estamos inserido imagens dessas mesmas no menu de opções ao lado de cada registro.

```
46  // Number of records to display on the screen
47  // Value of -1 lists all records in a table
48  $opts['inc'] = 15;
49
50  // Options you wish to give the users
51  // A - add,  C - change, P - copy, V - view, D - delete,
52  // F - filter, I - initial sort suppressed
53  $opts['options'] = 'AVC';
54
55  // Number of lines to display on multiple selection filters
56  $opts['multiple'] = '4';
57
58  // Navigation style: B - buttons (default), T - text links, G - graphic links
59  // Buttons position: U - up, D - down (default)
60  $opts['navigation'] = 'DG';
61
62  // Display special page elements
63  $opts['display'] = array(
```

Links PhpMyEdit :

site:[1]

Documentação:[2]

Fórum:[3]

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos/phpMyEdit"
Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos/DadaBik

De Wikibooks

< Aplicativos em PHP | Geradores de Aplicativos

7.4 - DadaBik

Mais um gerador de aplicativos para o PHP.

Site oficial - http://www.dadabik.org/

Ele usa o AdoDB como abstração de bancos de dados e segundo o site já foi bem testado nos SGBDs MySQL, PostgreSQL, Oracle e MS SQL Server.

Segundo o autor o DadaBik foi criado para ser customizável, tanto que para cada campo de tabela você pode escolher:

```
   * Se o campo será incluído ou não no formulário search/insert/update e
resultados da tabela
   * Rótulos dos campos
   * Formatos dos campos (numérico, alfabético, e-mail, url......)
   * O tipo para cada campo do form (select, date, text, rich text editor,
password......)
   * Os possíveis valores, também vindos de outra tabela (foreign key support)
   * e mais...
```

Veja mais detalhes e recursos no site oficial.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos/DadaBik"
Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos/PHPLibDev

De Wikibooks

< Aplicativos em PHP | Geradores de Aplicativos

7.5 - phpLibDev

Pequeno e experimental gerador de aplicativos que trabalha com PHP e PostgreSQL

É interessante observar as rotinas de metadados, que pegam informações diretamente do SGBD.

Existem duas versões: uma com funções do PHP acessanfo o PostgreSQL e outra usando a abstração PEAR/DB.

Projeto no Sourceforge - http://phplibdev.sourceforge.net/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos/PHPLibDev"
Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos/Web Form Generator

De Wikibooks

< Aplicativos em PHP | Geradores de Aplicativos

Web Form Generator

Gerador de Aplicativos para PHP com MySQL

Site oficial. Requer JRE 1.5 ou superior (http://developers.sun.com/downloads/top.jsp).

O Gerador é for Windows. Um stand alone em Java.

Muito flexível, pois permite, antes de gerar:

- Adicionar filtro para os campos (busca)
- editar os rótulos de cada campo
- alterar o tipo de controle dos campos do form (text, textarea, list, combo, radio, etc)
- alterar o tamanho dos campos (length) e o maxlength,
- adicionar validação como texto, e-mail ou número com mensagem de erro
- requeridos ou não
- marcar que campos aparecerão em cada tela: I (incluir), U (atualizar), D (excluir) e L (listar)

Traz recursos de paginação de resultados e estilo.

A lista de features no site

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos/Web_Form_Generator"

Page categories: PHP

Aplicativos em PHP/Geradores de Aplicativos/PHP Code Generator

De Wikibooks

< Aplicativos em PHP | Geradores de Aplicativos

Conteúdo

- 1 PHP Code Generator (com MySQL, PostgreSQL, Access e logo outros)
- 2 Site Oficial
- 3 Instalação
- 4 Configurações
- 5 Requisitos
- 6 Como Funciona
- 7 Documentação
- 8 Usando

PHP Code Generator (com MySQL, PostgreSQL, Access e logo outros)

Site Oficial

http://phpcg.sf.net

Do mesmo autor do PHPCodeGenie, Nilesh Dosooye.

O download da versão 0.2.1 (10/07/2007) em .tar tem uns 36.5MB, o .zip uns 4.9MB.

Instalação

Passo 1 : Descompacte no seu diretório web, por exemplo: /home/ribafs/htdocs/phpcg ou c:\xampp\htdocs\phpcg

Configurações

Passo 2 : Garanta permissão de escrita no diretório raiz e no subdiretório web\websites

Passo 3 : Caso exista um arquivo .htaccess no raiz do phpcg remova que outro será criado.

Para garantir o funcionamento tanto no Linux quanto no Windows edite o script index.php do raiz e altere a seguinte linha:

fwrite($fp, "php_value include_path ".$ROOT."\n");

Para:

```
fwrite($fp, "php_value include_path \"".$ROOT."\"\n");
```

Agora basta abrir no navegador

http://localhost/phpcg

Requisitos

Servidor:

Apache é recomendado, mas outros devem funcionar.

PHP:

Foi construido usando PHP 5.2 (não funciona com a versão 4).

SGBDs:

Qualquer um compatível com ADODB .

Obs.: Nomes de bancos, tabelas e campos não podem conter espaços

Como Funciona

http://phpcg.sourceforge.net/phpCG_framework.php

Documentação

http://phpcg.sourceforge.net/docs/framework_docs/index.php

No Linux, após gerar o site deve dar permissão ao usuário atual para que tenha acesso ao diretório gerado em web\websites.

Usando

Uso semelhante ao phpCodeGenie, apenas entre com os dados do banco e selecione o banco ou a tabela para gerar o aplicativo.

Para o Access apenas selecione Access na lista Database Type e indique o arquivo.

Testado com sucesso em MySQL, PostgreSQL e Access e como usa a abstração de dados ADODb teoricamente suporta todos os SGBDs suportados pela ADODb.

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/Joomla (Portais)

Conteúdo

- 1 Instalação
- 1.1 Requisitos do Joomla
- 1.2 Licença
- 1.3 Configurações do Banco
- 1.4 Nome do Portal
- 1.5 Acessar Site ou Administração
- 1.6 Portal Joomla Instalado
- 1.7 Portal Joomla (Frontend, ou seja, o site)
- 2 Joomla 1.0.12 Traduzido para Português do Brasil (Frontend e Backend)
- 3 Joomla 1.5 Traduzido para Português do Brasil (Frontend e Backend)
- 4 Administração do Joomla
- 4.1 Verificar lista de itens mais populares do site
- 4.2 Configurações do Site
- 4.3 Módulos
- 4.4 Seções e Categorias
- 4.5 Imagem
- 4.6 Banners
- 4.7 Enquete (Pools)
- 4.8 Contato
- 4.9 Mensagens entre Usuários
- 4.10 Enviar E-mail em Massa
- 4.11 Usuários
- 4.12 Estatísticas
- 4.13 Instalar e Desinstalar
- 5 Extensões Favoritas
- 5.1 Componentes
- 5.1.1 JoomlaXplore (Gerenciador de Arquivos)
- 5.1.2 PUArcade (jogos online)
- 5.1.3 Fun Games (jogos online)
- 5.1.4 EasyBook (Livro de Visitas)
- 5.1.5 RSGallery2 (Galeria de Imagens)
- 5.1.6 Biorritmo
- 5.1.7 JomComment (Comentários nas Seções)
- 5.1.8 MamboWiki (Adiciona mediawiki no Joomla)
- 5.1.9 Joomap (Mapa do Site)
- 5.2 Módulos
- 5.2.1 Bible Verse of The Day (VOTD)
- 5.2.2 javascript module
- 5.3 Mambots
- 5.3.1 JoomlaFCK
- 5.3.2 JoomlaTinyFCK (JTF)
- 5.4 Templates
- 5.4.1 Site com 1500 templates
- 5.5 Language
- 5.5.1 Português do Brasil
- 6 Personalizando Templates
- 7 Gerenciamento da Estrutura do Joomla
- 8 Entendendo o CSS do Joomla em 5 minutos
- 9 Referências

Instalação

Instalação do CMS Joomla 1.0.12

Download do Site Oficial - http://joomlacode.org/gf/project/joomla/frs/

Descompacte no seu DocumentRoot numa pasta, por exemplo, joomla1012 ou portal.

Então iniciar o apache e abrir o navegador com - http://localhost/joomla1012

Veja que será redirecionado para - http://127.0.0.1/joomla1012/installation/index.php

Requisitos do Joomla

O Joomla requer PHP e MySQL além de algumas configurações nesse sistema.

Observe as configurações e se seu sistema está pronto para a versão do Joomla que está instalando.

Em sistemas linux, caso não tenha criado o arquivo configuration.php e dado permissão de escrita, receberá o aviso:

```
" configuration.php         Unwriteable
You can still continue the install as the configuration will be displayed at the
end, just copy & paste this and upload."
```

Neste caso pode seguir em frente, mas fique bastante atento para

- Quando chegar ao final e o Joomla exibir o código para o configuration.php,

- Então copie o código,

- Crie o script configuration.php

- Abra o script e cole o código copiado

Caso seu register_globals esteja como "On" no php.ini, você será alertado sobre isso. Se puder altere para "Off".

Caso receba este aviso:

"Joomla! RG_EMULATION setting is `ON` instead of `OFF` in file globals.php `ON` by default for compatibility reasons"

Deverá alterar o arquivo globals.php, que se encontra no raiz do Joomla, mudando a linha:

define('RG_EMULATION', 1);

Alterando de 1 para 0.

A equipe do Joomla sabendo que muitos administradores de sites não tem acesso ao php.ini criou um emulador do mesmo, onde podemos emular o register_globals setando para Off.

Veja também as permissões de vários diretórios que requerem permissão de escrita. Caso esteja

usando Linux altere as permissões desses diretórios para 777 (sudo chmod 777 nomediretorio).

Please check the Official Joomla! Server Security post for more information.

Recommended Settings Check:

These settings are recommended for PHP in order to ensure full compatibility with Joomla!.

However, Joomla! will still operate if your settings do not quite match the recommended

Directive	Recommended	Actual
Safe Mode:	OFF:	OFF
Display Errors:	ON:	ON
File Uploads:	ON:	ON
Magic Quotes GPC:	ON:	ON
Magic Quotes Runtime:	OFF:	OFF
Register Globals:	OFF:	OFF
Output Buffering:	OFF:	OFF
Session auto start:	OFF:	OFF
Register Globals Emulation:	OFF:	ON

Directory and File Permissions Check:

In order for Joomla! to function correctly it needs to be able to access or write to certain files or directories.

If you see "Unwriteable" you need to change the permissions on the file or directory to allow Joomla! to write to it.

administrator/backups/	Unwriteable
administrator/components/	Unwriteable
administrator/modules/	Unwriteable
administrator/templates/	Unwriteable
cache/	Unwriteable
components/	Unwriteable
images/	Unwriteable
images/banners/	Unwriteable
images/stories/	Unwriteable
language/	Unwriteable
mambots/	Unwriteable
mambots/content/	Unwriteable
mambots/editors/	Unwriteable
mambots/editors-xtd/	Unwriteable
mambots/search/	Unwriteable
mambots/system/	Unwriteable
media/	Unwriteable
modules/	Unwriteable
templates/	Unwriteable

Ao chegar nessa tela, altere as permissões e altere o globals.php.

pre-installation check

Next >>

Check Again

Joomla! 1.0.12 Stable [Sunfire] 25 December 2006 01:00 UTC

Required Settings Check:

If any of these items are highlighted in red then please take actions to correct them.

Failure to do so could lead to your Joomla! installation not functioning correctly.

PHP version >= 4.1.0	**Yes**
- zlib compression support	**Available**
- XML support	**Available**
- MySQL support	**Available**
configuration.php	**Writeable**
Session save path **/tmp**	**Writeable**

Recommended Settings Check:

These settings are recommended for PHP in order to ensure full compatibility with Joomla!.

However, Joomla! will still operate if your settings do not quite match the recommended

Directive	Recommended	Actual
Safe Mode:	**OFF:**	**OFF**
Display Errors:	**ON:**	**ON**
File Uploads:	**ON:**	**ON**
Magic Quotes GPC:	**ON:**	**ON**
Magic Quotes Runtime:	**OFF:**	**OFF**
Register Globals:	**OFF:**	**OFF**
Output Buffering:	**OFF:**	**OFF**
Session auto start:	**OFF:**	**OFF**
Register Globals Emulation:	**OFF:**	**OFF**

Recommended Settings Check:

These settings are recommended for PHP in order to ensure full compatibility with Joomla!.

However, Joomla! will still operate if your settings do not quite match the recommended

Directive	Recommended	Actual
Safe Mode:	OFF:	OFF
Display Errors:	ON:	ON
File Uploads:	ON:	ON
Magic Quotes GPC:	ON:	ON
Magic Quotes Runtime:	OFF:	OFF
Register Globals:	OFF:	OFF
Output Buffering:	OFF:	OFF
Session auto start:	OFF:	OFF
Register Globals Emulation:	OFF:	OFF

Directory and File Permissions Check:

In order for Joomla! to function correctly it needs to be able to access or write to certain files or directories.

If you see "Unwriteable" you need to change the permissions on the file or directory to allow Joomla! to write to it.

administrator/backups/	Writeable
administrator/components/	Writeable
administrator/modules/	Writeable
administrator/templates/	Writeable
cache/	Writeable
components/	Writeable
images/	Writeable
images/banners/	Writeable
images/stories/	Writeable
language/	Writeable
mambots/	Writeable
mambots/content/	Writeable
mambots/editors/	Writeable
mambots/editors-xtd/	Writeable
mambots/search/	Writeable
mambots/system/	Writeable
media/	Writeable
modules/	Writeable
templates/	Writeable

Então tecle F5 ou clique em Check Again para atualizar.

Somente quando não tiver mais pendências siga em frente. Algumas pendências não impedem de seguir em frente.

Então clique em Next.

Licença

Receberá então a tela da Licença. Verifique o conteúdo ou simplesmente clique em Next.

Configurações do Banco

Aqui entre com os dados do banco de dados. Veja como exemplo os da imagem abaixo. Somnete no caso de já existir o banco e queira sobrescrever o existente, então marque a caixa "Drop Existing Tables".

Então clique em Next e confirme.

Nome do Portal

Entre com o nome do site e clique em Next

Acessar Site ou Administração

Apenas entre com seu e-mail e a senha para o usuario administrador. É recomendável trocar a senha default sob pena de você não lembrar da mesma, que é uma senha aleatória. Lembre que com isso irá acessar a seção de administração do site com login "admin" e esta senha.

Assim:

http://localhost/joomla1012/administrator

Após fazer o login altere o nome do usuário administrador.

Clique em Next.

Portal Joomla Instalado

Aqui é obrigatório (por segurança) que remova o diretório "installation" para poder acessar o site.

Nesta fase também, se no Linux, quando não deu permissão de escrita ao configuration.php, aparece o código para esse script. É quando deve copiar e colar no arquivo que deve criar (configuration.php).

Então acesse o site clicando em View Site ou a seção de administração clicando em Administration.

Portal Joomla (Frontend, ou seja, o site)

Joomla 1.0.12 Traduzido para Português do Brasil (Frontend e Backend)

Aqui - http://www.joomlaclube.org/index.php?option=com_content&task=view&id=34&Itemid=261

Ou aqui - http://sdsites.com.br/downloads/joomla_versao_PT_BR.zip

Joomla 1.5 Traduzido para Português do Brasil (Frontend e Backend)

http://www.joomlaclube.org/index.php?option=com_content&task=view&id=35&Itemid=261

Administração do Joomla

A adminsitração do Joomla é feita num outro site, http://localhost/joomla1012/admin

Login dos Administradores para backend

Painel de Controle da seção de administração do Joomla

Alguns recursos da área administrativa

- Usuários logados no portal

Veja que após o login como usuário super administrador você verá quais usuários estão logados neste momento e poderá inclusive derrubar (efetuar logout) algum usuário, se desejar, apenas clicando no X vermelho à direita do usuário, estando na aba Logado.

Verificar lista de itens mais populares do site

Na aba Popular aparecem as seções mais visitadas do site. Veja que existem outras abas que são atalhos para as principais funções encontradas nos menus e ícones.

Configurações do Site

Através do menu Site - Global Configuration podemos alterar diversas configurações do portal. Somente o super administrador poderá fazer essas alterações. Somente o usuário superadministrador tem acesso à esse item.

Aqui podemos trocr o editor HTML default

Aqui podemos trocr o nome do Site

Aqui podemos deixar o site offline (para manutenção)

Aqui podemos trocr o e-mail do contato configurado em Contato

Aqui podemos habilitar as estatísticas, para poder habilitar o contador (módulo Statistics).

E muitas outras configurações úteis.

Obs.: Alterações nesse item requerem permissão de escrita no arquivo configuration.php (do raiz do Joomla). Em termos de segurança é bom liberar quando for alterar e bloquear após as alterações. Para isso marque o checkbox "Override write protection while saving" antes de salvar.

Módulos

Após instalar o Joomla e instalar o idioma português do Brasil, então devemos fazer várias alterações manualmente: nomes dos módulos, nomes de menus, etc.

Ao editar um módulos podemos alterar a posição em que ele ficará na página: direita, esquerda, após qual módulo existente, etc. Também podemos configurar o acesso ao mesmo: público, registrado e especial (somente aqueles com maiores privilégios). Podemos e devemos alterar o título além de outras propriedades.

Seções e Categorias

Recursos muito úteis. Com eles podemos deixar a índex do site bem enxuta: somente com seções, que abrirão categorias, que abrirão itens de conteúdo.

```
seção M
    categoria A
              Item de Conteúdo 1
              Item de Conteúdo 2
              Item de Conteúdo 3
    categoria B
              Item de Conteúdo 1
              Item de Conteúdo 2
    categoria C
              Item de Conteúdo 1
Seção N
    categoria J
              Item de Conteúdo 1
              Item de Conteúdo 2
              Item de Conteúdo 3
    categoria K
              Item de Conteúdo 1
              Item de Conteúdo 2
    categoria L
              Item de Conteúdo 1
```

Imagem

Existe um recurso muito útil para enviar imagens por upload e outras mídias e também as administrar, que é o Midia Manager.

Sempre que quiser enviar uma imagem abra o Midia Manager, clique em Arquivo, selecione o arquivo e então clique em Enviar. Também podemos criar diretórios e mudar de diretórios.

Após enviar um arquivo para lá então podemos usar o editor HTML para adicionar a imagem a um artigo, por exemplo.

Banners

Sempre fique atento para o tamanho que se adapte corretamente ao template em que vai usar o banner.

Veja que pode zerar o contador de cliques para banner existente.

Pode controlar o período em que o banner será exibido.

Pode cadastrar o cliente e outras. O banner é um componente.

Enquete (Pools)

Personalize a enquete atual. Também pode adicionar tantas quantas queira. Uma enquete é um componente.

Contato

É importante configurar o contato logo que mande o site para o ar, adicionando os dados corretos do contato, e-mail, endereço (se for o caso), etc.

Mensagens entre Usuários

Recurso especial do Joomla, especialmente quando o site estiver sendo construído ou mantido por mais de um usuário. Neste caso os usuários podem trocar mensagens entre si. A área de mensagens, onde pode visualizar se tem ou não mensagem para você fica acima e à direita: um ícone de um envelope amarelo com uma seta curva e verde claro abaixo dele. Quando existir mensagem nova para você aparecerá o número de mensagens novas à esquerda do ícone. Ao clicar no ícone poderá ler as mensagens, responder, enviar nova ou excluir as lidas, além da opção de busca existente.

Enviar E-mail em Massa

Enviar um e-mail para todos os usuários registrados ou para grupos de usuários. Somente o super administrador tem este privilégio.

Usuários

O Joomla tem dois grupos de usuários, os do frontend (que tem acesso ao site) e os do backend (que tem acesso ao site administrativo).

```
Frontend
  public - (tem privilégio apenas de ver conteúdo liberado como públic)
    registered - (estes visualizam tudo que é public e também aqueles conteúdos
para usuários
      registrados. Quando logado pode enviar links)
      author - (estes tem os privilégios dos anteriores e também tem o de enviar
notícias e editar as
        próprias notícias)
        editor - (privilégios dos anteriores mais os de editar conteúdo de todos
do seu grupo)
          publisher (enviar, editar e publicar qualquer conteúdo no frontend.
Veja nenhum dos anteriores pode publicar seu conteúdo).

Backend
  public
    manager - este grupo tem acesso à criação de conteúdo e outras informações
do sistema
      administrator - este tem acesso à maioria das funções administrativas
        super administrator - este tem acesso à todas as funções administrativas
```

Estatísticas

Caso queira habilitar o contador de visitas, deverá ir em no menu Site - Global Configuration, aba Statistics e habilitar o item Statistics. Somente o super administrador tem este privilégio.

Então edite o módulo Statistics, no menu Modules - Site Modules. Então ative apenas o Contador.

Instalar e Desinstalar

Instalar e desinstalar extensões no Joomla é algo extremamente prático, usa-se apenas o mouse instalando os arquivos compactados.

Atente apenas para os tipos de extensões:

- **módulos** (estes ao serem instalados e publicados aparecem automaticamente no site. ao serem instalados estão despublicadso). Os módulos ao serem instalados permanecem despublicados.

- **componentes** normalmente requerem a criação de um item de menu que os clame ou de um link.

- **template** - transformam a aparência do site por inteiro. Após instalado para alterar o template do site vá no menu Site - Template manager - Site Tempaltes.

- **mambot** - geralmente são editores HTML. Após instalado vá no menu Site - Global Configuration e Default WYSIWYG Editor altere o editor. Somente o super administrador tem este privilégio

- **language** - instalar novo idioma.

Existem muitos outros recursos e a cada versão nova aparecem ainda outros novos recursos.

Estude, pesquise e compartilhe suas descobertas com outros por aqui.

Extensões Favoritas

Componentes

JoomlaXplore (Gerenciador de Arquivos)

Gerenciador de arquivos com muitos recursos úteis.

http://joomlacode.org/gf/project/joomlaxplorer/frs/

PUArcade (jogos online)

Jogos Clássicos: snake, asteroids, siga-me, space invader, pac-man, tetris e frogger.

http://www.pragmaticutopia.com/

Fun Games (jogos online)

407 Games free dos mais variados tipos.

http://www.freegaming.de/upload/com_fungames.zip

EasyBook (Livro de Visitas)

Bom livro de visitas com controle de spam.

http://joomlacode.org/gf/project/easyjoomla/frs/?action=FrsReleaseBrowse&frs_package_id=225

RSGallery2 (Galeria de Imagens)

Ótima galeria de imagens em português (quase tudo), com opção para envio de arquivos compactados, com galeria de imagens aleatórias, etc.

http://rsgallery2.net/

Biorritmo

Exibe gráfigo com biorritmo.

http://www.alikonweb.it/451/

JomComment (Comentários nas Seções)

Ótimo componente que permite adição de comentários nas seções. Veja uma amostra aqui

http://www.azrul.com/products/joomla_comment_system.html

MamboWiki (Adiciona mediawiki no Joomla)

Insere de fato o mediawiki no Joomla. Muito bom.

http://www.lyquidity.com/mambo/index.php?option=com_download&Itemid=180

Joomap (Mapa do Site)

Simples mapa de site.

http://www.koder.de/joomap.html

Módulos

Bible Verse of The Day (VOTD)

Exibe um versículo da Bíblia a cada entrada no site.

http://downloads.korske.com/files/mod_votd.zip

Bíblia Completa Online em PHP e MySQL

Existe uma bíblia completa (antigo e novo testamento) em PHP com MySQL que pode ser instalada e usada confortavelmente via wrapper no Joomla.

Veja aqui: http://br.groups.yahoo.com/group/bibliaonlinephp/

Cadastre-se gratuitamente na lista para fazer o download.

javascript module

Este módulo serve para adicionar código em JavaScript ou HTML, como por exemplo o AdSense do Google.

http://www.joomlastop.com/content/view/28/47/

Mambots

JoomlaFCK

Editor baseado no FCKEditor.

http://joomlacode.org/gf/project/joomlafck/frs/

JoomlaTinyFCK (JTF)

Editor tendo como base também o FCK.

http://www.hackjoom.web.id/JoomlaTinyFCK-mambot.html

Obs.: O editor default, que vem junto ao Joomla é muito bom.

Templates

Site com 1500 templates

http://www.joomlaos.de/joomla_template_galerie.html

Language

Português do Brasil

Para Joomla 1.0.12

http://www.geraldosales.com.br/portal/joomla/brazilian1.0.12.01.zip

Personalizando Templates

Podemos criar e personalizar templates para que atendam melhor nossas necessidades. Isso é uma das grandes vantagens do software open source.

Vou mostrar uma pequena personalização para os templates do Joomla ou Mambo, que pode ser aplicada a qualquer tempalte.

O objetivo é ler um arquivo TXT que contém uma coleção de pensamentos selecionados e ler uma linha a cada carga do site, no caso irá mostrar um pensamento a cada vez que o visitante vier ao site.

Veja o código alterado da index.php de um template:

```
// Adicionei a função abaixo no início da index.php do templates
// lembrar que o arquivo pensamentos.txt encontra-se no raiz do template
```

```
<?php
function pensamento_aleatorio($arquivo){
$fn = "$GLOBALS[mosConfig_live_site]/templates/$GLOBALS[cur_template]/$arquivo";
$f_contents = file ($fn);
srand ((double)microtime()*1000000);
$linha_aleatoria = $f_contents[ rand (0, (count ($f_contents) - 1)) ];
print $linha_aleatoria;
}
?>

<?php defined( "_VALID_MOS" ) or die( "Direct Access to this location is not
allowed." );$iso = split( '=', _ISO );
echo '<?xml version="1.0" encoding="'. $iso[1] .'"?' .'>';?>
//...

// Inseri a chamada da função após este trecho da index.php do template free
247portal-b-green
/*
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="11" height="25" background="<?php echo
$mosConfig_live_site;?>/templates/247portal-b-green/images/shadowl.jpg"><div>
</div></td>
<td height="25" bgcolor="#F1F1F1" style="border-bottom: 1px solid #999999;
border-top: 5px solid #FFFFFF;"><?php mosPathWay(); ?></td>
<td height="25" align="right" bgcolor="#F1F1F1" style="border-bottom: 1px solid
#999999; border-top: 5px solid #FFFFFF;"><div class="date">
<?php echo mosCurrentDate(); ?></div></td>
<td width="11" height="25" align="right" background="<?php echo
$mosConfig_live_site;?>/templates/247portal-b-green/images/shadowr.jpg"> </td>
</tr>
</table>
// Aqui a chamada da função:
<center><font color="blue"><b><i><?php
pensamento_aleatorio("./pensamentos.txt"); ?></i></b></font></center>
*/
?>
```

O arquivo pensamentos.txt pode ser criado ou alterado por você, mas se preferir um pronto faça o download deste abaixo e converta para .txt. Precisará do OpenOffice para abrir.

http://upload.wikimedia.org/wikibooks/pt/9/9b/Pensamentos.sxw

Gerenciamento da Estrutura do Joomla

http://www.joomlaclube.org/index.php?option=com_content&task=view&id=121&Itemid=427

Entendendo o CSS do Joomla em 5 minutos

http://www.joomlaclube.org/index.php?option=com_content&task=view&id=124&Itemid=425

Referências

Site Oficial do Joomla - http://www.joomla.org

Site com grande catálogo de extensões para Joomla - http://extensions.joomla.org/

Forum - http://forum.joomla.org/

Templates (mais de 1500 templates grátis) - http://www.joomlaos.de/joomla_template_galerie.html

Joomla Brasil – http://www.joomlabrasil.org

Joomla do Brasil – http://www.jooma.com.br

Joomla Clube – http://www.joomlaclube.org

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/Joomla_%28Portais%29"
Page categories: PHP

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/Mambo (Portais)

De Wikibooks

< Aplicativos em PHP | CMSs (Gerenciadores de Conteúdo)

8.2 - Mambo

O Mambo já foi o CMS mais premiado. Criado pela empresa Miro e aberto à comunidade com a licença GPL. Houve desentendimento entre a empresa e a comunidade fazendo surgir há pouco mais de um ano o Joomla.

Site para download - http://www.mamboserver.com/

Tutoriais em Português

- http://mambobrasil.org/

- http://ribafs.net/down/tut/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/Mambo_%28Portais%29"
Page categories: PHP

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/Drupal (Portais)

De Wikibooks

< Aplicativos em PHP | CMSs (Gerenciadores de Conteúdo)

Instalação

- Crie um banco de dados "drupal"
- Importe o script da pasta database
- Configure o script da pasta sites/default para os dados do SGBD
- Abra no browser com http://127.0.0.1/drupal-4.7.3/
- Então receberá:

Welcome to your new Drupal website!

Please follow these steps to set up and start using your website:

1.Create your administrator account To begin, create the first account. This account will have full administration rights and will allow you to configure your website.

2.Configure your website Once logged in, visit the administration section, where you can customize and configure all aspects of your website.

3.Enable additional functionality Next, visit the module list and enable features which suit your specific needs. You can find additional modules in the Drupal modules download section.

4.Customize your website design To change the "look and feel" of your website, visit the themes section. You may choose from one of the included themes or download additional themes from the Drupal themes download section.

5.Start posting content Finally, you can create content for your website. This message will disappear once you have published your first post.

For more information, please refer to the Help section, or the online Drupal handbooks. You may also post at the Drupal forum, or view the wide range of other support options available."

Dica - clique com o botão direito sobre o primeiro link para abrir numa outra janela e ter os links originais sempre à disposição.

Após criar o user administrador:

- Clique em administer
- Depois em settings para configurar o CMS. Verifique e altere todas que julgar importantes.
- Então clique em modules para ver a lista de módulos ativos e ativar/desativar os desejados. Habilite o módulo locale para mudar para português do Brasil, por exemplo.
- Baixe o arquivo de tradução do site oficial e:
    - administer - localization - manage languages - import. Selecione o arquivo e em import into selecione Português.
- Altere a default para Português

- Módulos adicionais - http://drupal.org/project/modules
- Customizando o tema do site - http://127.0.0.1/drupal-4.7.3/?q=admin/themes
- Selecionando um tema - http://drupal.org/project/themes
- Começar a adicionar conteúdo - http://127.0.0.1/drupal-4.7.3/?q=node/add
- Ajuda - http://127.0.0.1/drupal-4.7.3/?q=admin/help, http://drupal.org/handbooks, http://drupal.org/forum, http://drupal.org/support, etc.
- Criando Conteúdo http://127.0.0.1/drupal-4.7.3/?q=node/add

Ativando/Desativando módulos

http://127.0.0.1/drupal-4.7.3/?q=admin/modules

Algumas sugestões.

- Verifique as respectivas explicações na seção para ver se concorda:

- aggregator, archiver, blog, blogapi, book, forum, pool, search, statistcs, tracker (ativar)

Caso tenha marcado os itens acima e acesse novamente: http://127.0.0.1/drupal-4.7.3/?q=node/add

agora verá algo como:

Principal

criar conteúdo

Escolha o ítem apropriado na lista:

enquete

Enquete é uma pesquisa de opinião de múltipla escolha onde os usuários podem votar em uma das opções apresentadas.

entrada de blog

Um blog é um diário ou jornal atualizado regularmente feito por contribuições individuais mostradas em ordem cronológica reversa. Um blog está estreitamente ligado ao autor, portanto cada usuário terá seu "próprio" blog.

matéria

Matérias são artigos em sua forma simples: elas possuem um título, uma chamada e um corpo, mas podem ser extendidas por outros módulos. O teaser é parte do corpo também. Matérias podem ser usadas como um blog pessoal ou para artigos de notícias.

página

Se você deseja adicionar uma página estática, como uma página de contato ou uma página "sobre", use a página.

página de livro

Um livro é um esforço de criação coletiva: os usuários podem contribuir escrevendo páginas para o livro, posicionando as páginas na ordem correta e revisando e modificando páginas previamente escritas. Assim, quando você tem alguma informação para compartilhar, quando leu uma página do livro que não gostou ou se achou que uma página poderia ser melhor escrita você pode fazer algo para mudar as coisas.

tópico de fórum

Criar um novo tópico para discussão nos fóruns.

Edita cada seção e crie seu portal.

Para Instalar com postgresql

Criar Usuário

create role drupal with nocreateuser nocreaterole nocreatedb encrypted password 'drupal';

Criar banco:

create database drupal with encoding 'unicode' owner drupal;

Dica: Caso esteja instalando em micro sem servidor de e-mail:

Execute este código:

print md5('seulogin'); //Exibirá o hash md5

Então abra o phpmyadmin com o banco do drupal, selecione a tabela user e visualize os registros edite o registro do seu login e substitua o hash md5

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/Drupal_%28Portais%29"
Page categories: PHP

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/Xoops (Portais)

De Wikibooks

< Aplicativos em PHP | CMSs (Gerenciadores de Conteúdo)

CMS para criação de Portais

O Xoops é um dos CMS mais populares.

Site oficial - http://www.xoops.org

Site oficial de suporte ao brasil - http://xoopsbr.org/

Download - http://www.xoops.org/modules/core/

Instalação - http://xoopsdocs.net/modules/docs/en/xu-002/online/index.html

A instalação é semelhante a de outros CMS.

Documentação - http://xoopsdocs.net/modules/docs/

Installing And Working With Xoops Under Ubuntu 6.10

http://www.howtoforge.com/xoops_ubuntu

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/Xoops_%28Portais%29"
Page categories: PHP

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/MediaWiki (wikis)

De Wikibooks

< Aplicativos em PHP | CMSs (Gerenciadores de Conteúdo)

O Mediawiki é o CMS que deu origem à Wikipedia, ao Wikibooks e a toda essa geração de CMS que trouxe a democracia ao conhecimento.

Conteúdo

- 1 Instalação do Mediawiki
- 2 Configurações de pós-instalação
- 3 Usando o MediaWiki
  - 3.1 Formatação de Texto
- 4 Criando Índice em HTML
- 5 Inserindo Imagens e Outros arquivos
  - 5.1 Outras opções para inserir imagem
- 6 Criando um Índice com itens e subitens
- 7 Esboço de página
- 8 Evitar Interpretação do Mediawiki
- 9 Seções
- 10 Comunicação entre Usuários registrados
- 11 Código Fonte
- 12 Wiki: ser ou não ser?
- 13 Extensão para Firefox da MediaWiki
- 14 Referências

Instalação do Mediawiki

Instalação no Linux Ubuntu (é semelhante a uma instalação no Windows ou em outro Linux).

Compatibilidade

Lembrando que o Mediawiki atual (1.10.0) funciona somente com o PHP5 ou superior. Caso precise instalar em algum servidor com o cPanel precisará de uma versão para PHP 4.3.x ou superior, no caso a versão 1.6.10 ou outra compatível.

- Baixar de http://www.mediawiki.org
- Fazer o download para o documentRoot no diretório "mediawiki"
- Dar permissão de escrita ao diretório config (chmod a+w config)
- Criar o usuário e o banco, tendo o usuário privilégios sobre o banco.
- Chamar no browser http://localhost/mediawiki
- Preencha devidamente os dados, atentando para os requeridos.
- Após confirmar deverá mover o arquivo config/LocalSettings.php para o raiz do mediawiki.
- Então acesse http://127.0.0.1/mediawiki/index.php

Para acessar como administrador clique em Entrar e faça o login.

Configurações de pós-instalação

Alterando o logotipo

O logotipo que aparece no canto superior esquerdo poderá ser alterado. Para isso, no arquivo LocalSettings.php, insira ou altere o valor da linha $wgLogo.

Exemplo:

$wgLogo = "/wiki/eriberto.png"

Lembre-se de fazer upload do novo arquivo LocalSettings.php e da imagem referente ao logotipo.

Alterando o rodapé

Para alterar o rodapé do Wiki (retirando o Disclaimer, por exemplo), edite o arquivo skins/MonoBook.php e remova as opções indesejadas constantes na linha $footerlinks. As opções existentes nesta linha podem ser vistas no FAQ da MediaWiki (veja How do I change the footer?).

Alterando o fuso horário

Para que o Wiki utilize o mesmo fuso horário do usuário, no momento de uma edição, adicione no final do arquivo LocalSettings.php, antes de ?>, a seguinte linha:

$wgLocalTZoffset = date("Z") / 60;

Alterando as mensagens de tela e menus

É possível alterar as mensagens que aparecem na tela e os nomes que aparecem nos menus. Para isso, edite o arquivo languages/messages/MessagesPt_br.php.

Permitindo o upload de imagens

Para permitir que as pessoas que editam o wiki façam o upload de arquivos, altere o arquivo LocalSettings.php e altere a linha:

$wgEnableUploads = false

para

$wgEnableUploads = true

Usando o MediaWiki

Para texto puro é semelhante aos editores de texto comuns e aos editores HTML, com exceção da quebra de linha que aqui é efetuada teclando-se duas vezes no Enter ao invés de uma.

Formatação de Texto

Negrito - Apenas selecione o texto e clique no botão B na barra do editor acima.

Alternativamente selecione o texto e insira ''' antes e após o texto selecionado (o mesmo efeito de clicar no **B**).

Itálico - Proceder de forma semelhante clicando no ***I***.

Título

Para formatar um texto como título clique no botão **Texto a negrito**A da barra do editor acima.

Indentação

Para indentar usa-se dois pontos ":"

```
:Primeiro nível
::Segundo
:::Terceiro
```

Aparecerá assim:

Primeiro nível
Segundo
Terceiro

Indentar com Pontos

```
*Primeiro nível
**Segundo
***Terceiro
```

Listas Numeradas

Listas Numeradas

```
#Primeiro

#Segundo

##SubdoSegundo

#Terceiro
```

Aparecerá:

1. Primeiro
2. Segundo
    1. SubdoSegundo
3. Terceiro

Quebra de Linha

Podemos usar a quebra de linha do HTML <br>.

Assim como podemos utilizar vários comandos do HTML, mas nem todos.

Caso se digite:

```
Este parágrafo está
em três
linhas.
```

O Mediawiki (Wikibooks) irá exibir o seguinte parágrafo:

Este parágrafo está em três linhas.

Para que seja de fato exibido em três linhas deverá inserir sempre uma linha em branco entre cada uma delas, assim:

Este parágrafo está

em três

linhas.

Ou usando <br>, assim:

```
Este parágrafo está<br>
em três<br>
linhas.<br>
```

Que aparecerá assim:

Este parágrafo está
em três
linhas.

Criando Índice em HTML

Caso queiramos criar automaticamente um índice da página que estamos criando, basta usar a tag título <H1>, <H2> ... <H6>. Edite esta página ou outra com este recurso e confira.

Somente funciona se a página contiver um título usado com o ícone A, ou seja ==Título==.

Link

Insira link selecionando a URL e clicando no botão com o globo da barra.

Links para Páginas Externas

[http://pt.wikibooks.org/ WikiBooks]

Abre colchete, URL, Descrição, espaço em branco e fecha colchete.

Se apenas digitarmos http://pt.wikibooks.org/ também aparecerá o link e o rótulo será a própria URL.

Criando Nova Página

- Suponha que esteja em http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos e queira criar uma nova página para um outro gerador ainda não conttemplado

- Apenas faça login (de preferência) e abra no navegador a seguinte URL:

http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Geradores_de_Aplicativos/MeuGerador

Irá receber a mensagem "(Não há atualmente nenhum texto nesta página)". Então apenas clique em Editar e insira o conteúdo para o gerador.

Inserindo Imagens e Outros arquivos

A lista de arquivos permitidos no Wikibooks é somente esta:

png, gif, jpg, jpeg, xcf, xls, pdf, mid, sxw, sxi, sxc, sxd, ogg, svg, djvu

```
- Clique no ícone para inserir imagem na barra ou simplesmente digite no texto
[[Imagem:nomearquivo.jpg]]
- Salve a página
- Clique no link gerado
- Faça o upload do arquivo
- Salve e volte ao texto contendo a imagem (atualize com F5)
- Clique sobre a imagem para exibi-la
```

```
- Clique em editar e digite uma das predefinições para imagem, como por exemplo:

{{GFDL}}
```

No caso das imagens, as predefinições a serem usadas para licença são:

```
    * {{Cc-by-sa-2.5}} para licença Creative Commons Atribuição -
Compartilhamento pela mesma licença (veja Predefinição:Cc-by-sa-2.5)
    * {{GFDL}} para licença pública GNU FDL (veja Predefinição:GFDL)
    * {{DP}} para imagens em domínio público. (veja Predefinição:DP)
```

Outras opções para inserir imagem

```
[[image:nomeimagem.png|thumb|rótulo]]
```

Exibirá a miniatura da imagem com o rótulo.

```
[[image:nomeimagem.png|left|thumb|50px|]]
```

Exibirá a miniatura da imagem com o rótulo e tamanho 50 pixel e alinhado pela esquerda.

Criando um Índice com itens e subitens

Faça como o exemplo abaixo:

```
#[[/Introdução/]] [[Imagem:3de8.png]]
##[[/Introdução/História|História]]
##[[/Introdução/Características e Recursos|Características e Recursos]]
##[[/Introdução/A quem se destina este livro|A quem se destina este livro]]
#[[/InteligênciaEmocional/]]
#[[/Instalação e configurações do ambiente/]]
##[[/Instalação e configurações do ambiente/Via Xampp|Via Xampp]]
##[[/Instalação e configurações do ambiente/Individualmente|Individualmente]]
#[[/Editores e IDEs para PHP, CSS, JavaScript e HTML/]]
##[[/Editores e IDEs para PHP, CSS, JavaScript e HTML/PHPDesigner|PHPDesigner]]
##[[/Editores e IDEs para PHP, CSS, JavaScript e HTML/Kate|Kate]]
```

Detalhe importante: ao criar um índice assim bastará clicar em qualquer um dos capítulos ou itens para que a página seja criada em branco e possamos adicionar conteúdo.

Obs.: Veja que na primeira linha aparece uma pequena imagem ao lado direito. O wikibooks tem diversas dessas imagens. Veja na Biblioteca.

Para que apareça uma imagem representando o estágio atual do texto de um capítulo ou item use:

```
[[Imagem:3de8.png]]
```

à direita do item.

Esboço de página

Quando apenas iniciamos uma página podemos chamar a atenção do leitor e até solicitar sua colaboração com a seguinte tag:

{{stub}}

Que gera o seguinte aviso:

Esta página é somente um esboço.
Expandindo-a *você ajudará a melhorar o Wikilivros.*

Evitar Interpretação do Mediawiki

Uma forma de evitar que o media wiki interprte o que escreve é usando a tag <pre>... </pre>.

Para que de fato nunca interprete usar a tag <nowiki>... </nowiki>.

Para conhecer mais detalhes do uso do Mediawiki edite qualquer página e observe o conteúdo atentamente.

Seções

```
==Secão==
===Subseção===
====Sub-subseção====
```

Comunicação entre Usuários registrados

Os usuários registrados do Wikibooks (mwdiawiki) comunicam-se através de seus forums, cujo link "discussão" e "minha discussão" acima. Este último aparece somente quando estamos logados.

- No caso devemos acessar a área do usuário com quem queremos conversar

- Clicamos em discussão

- E clicamos no link do assunto já aberto.

- Caso queiramos abrir um novo tópico clicamos no sinal de + e criamos o tópico

Sempre ao final das mensagens devemos nos lembrar de assinar a mensagem.

Assinar Mensagens

Assina-se a mensagem com ~~~ (3 tils, que irão exibir o login do usuário)

ou com ~~~~ que mostrarão o login e a data completa.

Código Fonte

Ao chegar a um site wiki e ao invés de editar aparecer Exibir Código Fonte é porque o administrador impediu a edição por algum motivo.

Wiki: ser ou não ser?

Ótimo artigo do Carlos Nepomuceno sobre os valores do wiki - http://www.dicas-l.com.br/conhecimento_em_rede/conhecimento_em_rede_20070510.php

Extensão para Firefox da MediaWiki

Após ter se acostumado com a sintaxe do mediawiki é interessante e produtivo usar algo como essa extensão que ajuda na criação e edição das páginas.

Site para instalação - https://addons.mozilla.org/en-US/firefox/addon/1662

Referências

- Tutorial de Mediawiki em português do Eriberto - http://www.eriberto.pro.br/wiki/index.php?title=Instala%C3%A7%C3%A3o_do_MediaWiki_em_sites_pessoais_em_WebHostings

-

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/MediaWiki_%28wikis%29"
Page categories: Esboço | PHP

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/WordPress (blogs)

De Wikibooks

< Aplicativos em PHP | CMSs (Gerenciadores de Conteúdo)

8.6 - Wordpress

Instalação

Instalação do CMS Wordpress

Página oficial - http://wordpress.org/

Download - http://wordpress.org/download/

Documentação - http://codex.wordpress.org/Main_Page

Existem dois modos de instalar o Wordpress. Um deles é utilizando o painel de administração do seu provedor de serviços(host), o outro é instalando manualmente o sistema através de acesso FTP. Como as instalações do tipo "clicou-instalou" devem seguir os procedimentos estabelecidos pelo host, iremos abordar a instalação via FTP.

Requerimentos

- Browser de sua preferência;

- Acesso ao disco rígido em que o sistema será instalado;

- Editor de textos;

- PHP versão 4.2 ou superior;

- MYSql versão 4.0 ou superior.

Estando tudo de acordo seguimos para a instalação do sistema.

- Faça o download do pacote direto do site oficial;

- Descompacte o arquivo no diretório que quiser. Isto irá descompactar um diretório de nome *wordpress*;

- Entre neste diretório e edite o arquivo *wp-config-sample.php*. Para isso, abra o arquivo e você verá, entre outras, as três linhas seguintes:

```
define('DB_NAME', 'wordpress');
define('DB_USER', 'username';
define('DB_PASSWORD', 'password');
```

Você deve substituir os termos em negrito de acordo com o que está abaixo:

DB_NAME: É o nome do banco de dados (que você criou ou ainda criará);

DB_USER: O nome de usuário do banco de dados;

DB_PASSWORD: A senha de usuário do banco de dados;

- Salve o arquivo como wp-config.php

- Carregue todos os arquivos desse diretório no host que você está utilizando;

- Crie o banco de dados apontado no arquivo acima no host, caso ainda não tenha criado.

- Ao apontar o browser para a pasta em que o wordpress foi colocado uma mensagem pedindo para que você acesse o arquivo "install.php" será exibida. Clique no link que liga a este arquivo e a instalação terá início.

- Clique em First Step

- Dê um título ao seu WebLog e entre com o e-mail do admin.

- Então clique em Continue to Second Step

- Agora serão criadas as tabelas e populadas com informações. Anote o login (admin) e a senha indicada para o acesso. Aqui indicou "admin" e "94db5c".

Então clique no link

- http://127.0.0.1/wordpress/wp-login.php para fazer o acesso, fornecendo os dados anotados.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/WordPress_%28blogs%29"
Page categories: PHP

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/eGroupWare (colaboração)

De Wikibooks

< Aplicativos em PHP | CMSs (Gerenciadores de Conteúdo)

8.7 - eGroupWare (Colaboração)

O eGroupWare é um software de colaboração escrito em PHP com suporte aos SGBDs MySQL, PostgreSQL, Oracle, SAP/ODBC, SQL Sevrver

Instalação em distribuições baseadas no Debian

apt-get install egroupware

apt-get install egroupware-manual

Instalação no Windows

- Download da versão .zip de www.egroupware.org

40MB a versão 1.2-104 e após descompactado ocupa

- Descompacte no diretório web

- Abra no browser com

`http://127.0.0.1/egroupware`

- Selecione o idioma (Brazil)

- Clique no link Executar Testes de Instalação

- Corrija os erros (vermelho) e avisos (amarelo) e repita a checagem.

Então clique em Prosseguir para o Cabeçalho Admin.

- Veja que ele já tenta detectar os SGBDs instalados.

Preencha as informações que criação o script header.inc.php

Lembre de criar antes o banco de dados (o egroupware pode criar na instalação), o usuário para o egroupware e entrar em Usuário da base de dados inclusive senha ou usar um usuário existente.

Como também o usuário de configuração.

e clique em Configuração de Escrita

- Clique em Continuar
- Entre com login e senha para definir a conexão com o Administrador e clique em Login.
- Clique em Verificar novamente a instalação
- Clique em "Instalar" todos os aplicativos.
- Clique em Verificar novamente a instalação
- Clique em Editar configuração atual (caso apareça, e-mail por exemplo) e corrija as configurações.
- Clique em Criar conta do administrador

Entre com os dados para o administrador e clique em Salvar.

Quando tudo estiver OK clique em Desconectar e acesse:

```
http://localhost/egroupware
```

Faça o login como usuário de configuração e gerencie o eGroupWare, crie compromissos, usuários, tarefas, projetos, etc.

Muito bom este CMS. Após instalar confira seus recursos e o personalize à vontade.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/eGroupWare_%28colabora%C3%A7%C3%A3o%29"
Page categories: PHP

Aplicativos em PHP/CMSs (Gerenciadores de Conteúdo)/Moodle (eLearning)

De Wikibooks

< Aplicativos em PHP | CMSs (Gerenciadores de Conteúdo)

8.8 - Moodle (eLearning)

Moodle Gerenciador de Conteúdos para Cursos Online

Este tutorial foi baseado em uma instalação do Moodle 1.6.1 for Windows.

Com as devidas alterações pode ser adaptado para uma instalação no Linux.

1 - Download - www.moodle.org

Baixar sempre a última versão estável para seu sistema operacional.

2 - Descompactar no diretório web.

Se no Linux ou similar crie o diretório moodledata no mesmo nível do diretório do moodle, com permissão de escrita.

3 - Criar o banco para o moodle (moodle). Atualmente suporta MySQL e PostgreSQL.

4 - Abrir o browser com http://127.0.0.1/moodle/

5 - Selecionar o idioma (Português - Brasil) e Next

6 - Se tudo OK clique em Próximo. Caso contrário corrija as pendências.

7 - Na tela Instalação apenas clique em Próximo

8 - Na tela de configuração do SGBD, entre com os dados e clique em Próximo.

9 - Se tudo OK clique em Próximo

10 - Se conectado à Internet e sugerido "Baixar o Pacote de Idioma", então faça o download e depois clique em Próximo.

11 - Caso o Moodle não consiga criar o arquivo config.php no raiz do diretório moodle, então crie manualmente o arquivo e dê permisão de escrita ao mesmo. Ou então faca o download sugerido no diretório do moodle.

12 - GPL License. Ao final clique em Yes.

13 - Setting up database. Ao final clique em Continue

14 - Administration. Clique em Continue.

15 - Variables. Faça as devidas alterações e ao final clique em Save changes.

16 - Setting up module tables. Ao final clique em Continue.

17 - Setting up plugin tables. Ao final clique em Continue.

18 - Upgrading database. Ao final clique em Continue.

19 - Upgrading database. Ao final clique em Continue.

20 - Setting up block tables. Ao final clique em Continue.

21 - Setting up plugin tables. Ao final clique em Continue.

22 - Site settings.

Entre com as informações para a criação do site e faça as configurações que desejar. Ao final clique em Save changes.

23 - Configurar conta do administrador. Entre com os dados do usuário administrador e clique em Update profile.

24 - Concluído. Estamos no site, logados como admin.

25 - Adicionar Nova Categoria

- Administração (Admin...) - Cursos (Courses) - Administrar Cursos (Manage courses)

Mais detalhes em - http://docs.moodle.org/en/Installing_Moodle

Cópia local em - lang/en_utf8/help/install.html (or your language)

Curso sobre o Moodle em http://cursos.cdtc.org.br e http://comunidade.cdtc.org.br

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_%28Gerenciadores_de_Conte%C3%BAdo%29/Moodle_%28eLearning%29"
Page categories: PHP

Aplicativos em PHP/FrameWorks/P4A

De Wikibooks

< Aplicativos em PHP | FrameWorks

Conteúdo

- 1 P4A – PHP for Application
- 2 RAD
- 3 Framework
- 4 Elementos do P4A
  - 4.1 Masks
  - 4.2 Widgets
- 5 Características
  - 5.1 Busca
  - 5.2 Widgets
- 6 Desenvolvendo Aplicativos com o P4A
  - 6.1 Exemplo de Estrutura Mínima de uma Aplicação em P4A
  - 6.2 Primeiro Exemplo – Olá Mundo
  - 6.3 Segundo Exemplo – Acessando Banco de Dados
  - 6.4 Exemplo Com 3 Tabelas
- 7 Dicas Extras

P4A – PHP for Application

Site oficial - http://p4a.sf.net

P4A (PHP For Applications) é um framework em PHP tipo RAD e orientado a objetos para construção event-driven de aplicações web. Usa tableless HTML, suporte às teclas de atalho, point&click gerador de aplicativos, reconhecimento automático de datos, UTF-8, i18n/l10n, integração com a biblioteca PEAR.

De autoria do italiano Fabrizio Balliano.

Segundo a Wikipédia (http://pt.wikipedia.org):

RAD

Rapid Application Development (RAD), também conhecido como Desenvolvimento Rápido de Aplicação, é um modelo de processo de desenvolvimento de software iterativo e incremental que enfatiza um ciclo de desenvolvimento extremamente curto (entre 60 e 90 dias). O termo foi registrado por James Martin em 1991 e tem substituído gradativamente o termo de prototipação rápida que já foi muito utilizada no passado

Framework

No desenvolvimento do software, um framework é uma estrutura de suporte definida para que outro projeto de software possa ser organizado e desenvolvido. Tipicamente, um framework pode incluir programas de apoio, bibliotecas de código, linguagens de script e outros softwares para ajudar a desenvolver e juntar diferentes componentes de um projeto.

Elementos do P4A

Masks

É o objeto básico de interfaces, que contem todos os widgets e geralmente exibe objetos.

Localizado em /p4a/objects/mask.php (linha 45)

```
P4A_Object
   |
   --P4A_Mask
```

Widgets

São os elementos GUI prontos que podemos usar para compor nossas máscaras (masks). Classe base para objetos que permitem a usuários interagirem com a aplicação: botões, menus, .

Localizado em /p4a/objects/widget.php (linha 48)

```
P4A_Object
   |
   --P4A_Widget
```

O P4A é um projeto abrigado no SourceForge desde fevereiro de 2003, que chegou ao 12o. lugar no ranking geral dos mais de 100.000 projetos do SourceForge, tendo ganhado alguns prêmios neste repositório. Atualmente fica em primeiro lugar entre os frameworks de PHP no SourceForge.net.

Já conta com locales para uns 160 países e tradução para uns 60, inclusive o Brasil (pt_BR).

P4A é um framework PHP, um software contendo bibliotecas, módulos e controles (widgets) usados para a construção de aplicações que devem ser usadas com um web browser conectado à Internet. Com P4A você estará habilitado a desenvolver e escrever código de aplicações web como você faria com as ferramentas RAD mais evoluídas. O resultado deverá ser uma aplicação que o usuário deverá usar pensando que está usando uma aplicação cliente clássica (tipo desktop).

Você não deve se preocupar com a criação da interface gráfica para seus projetos, porque P4A já tem um tema gráfico cliente agradável.

Características

- Escrito em PHP (www.php.net), compatível com as versões PHP 4.3.x e 5.x.x e Apache 1.3.x e 2.0.x em Linux e Windows

- Integrado com a biblioteca Pear/DB (http://pear.php.net) e com isso suporta todos os bancos da suportados pela PEAD/DB: PostgreSQL, MySQL, SQL Server, Oracle e outros.

- Lógica e Design separados pelo template (http://pear.php.net/package/HTML_Template_Flexy)

Busca

Outras informações sobre o Projeto P4A:

```
- Home Page – http://p4a.sourceforge.net
- Downloads - http://sourceforge.net/project/showfiles.php?group_id=98294
- Demo online - http://p4a.sourceforge.net/demo
- Forum em italiano - http://sourceforge.net/forum/forum.php?forum_id=473897
- Forum em inglês - http://sourceforge.net/forum/forum.php?forum_id=340765
- Quem está usando - http://p4a.sourceforge.net/who-s-using-p4a
- Referência de Código - http://p4a.sourceforge.net/code-reference
- Tutoriais - http://p4a.sourceforge.net/tutorials
- Widgets - http://p4a.sourceforge.net/widgets
- Icon Packs - http://p4a.sourceforge.net/icons-packs
- Contribuições - http://p4a.sourceforge.net/info-about-contribs (Para instalar
uma contribuição, basta descompactar no diretória da sua aplicação.
Após instalar o contrib todos os objetos necessários ao P4A serão
automaticamente incluídos na sua aplicação).
- Protótipo de Contribuição - http://p4a.sourceforge.net/contrib-prototype
(protótipo vazio para ajudar pessoas a criarem uma nova contribuição).
```

```
- Logo mais abaixo da seção anterior (no site oficial) outros aplicativos de
terceiros úteis.
```

Widgets

São os elementos GUI prontos que podemos usar para compor nossas máscaras (masks). Classe base para objetos que permitem a usuários interagirem com a aplicação: botões, menus, .

Esta lista (dos principais) é aninhada para reproduzir a herança das classes. Todos estes objetos extendem a classe P4A_Widget. Veja a referência de código para detalhes.

```
    * Button
    * Canvas
    * Field
    * Frame
          o Fieldset
    * Menu
    * Message
    * Table
    * Tab pane
    * Toolbar
          o Actions toolbar
          o Navigation toolbar
          o Simple toolbar
          o Standard toolbar
          o Quit toolbar
```

Desenvolvendo Aplicativos com o P4A

Pré-Requisitos:

- Sistemas Operacionais Linux ou Windows

- PHP 4.3.x/4.4.x/5.x.x com Apache 1.3.x/2.0.x

- A biblioteca PEAR/DB

- Um banco de dados suportado pela bibliteca acima, caso o seu aplicativo use banco

Quando instalamos o Xampp (http://xampp.sf.net), ele provê todos os requisitos do P4A, inclusive tem versão para ambos os sistemas operacionais.

Quando descompactamos o p4a ele cria a seguinte estrutura de diretórios:

```
p4a/applications
p4a/docs
p4a/icons
p4a/libraries
p4a/p4a
p4a/themes
p4a/index.php
```
p4a/p4a.php

Os aplicativos criados devem ficar na pasta “applications”.

Na pasta “docs” está toda a referência do P4A.

Em “icons” estão os ícones utilizados, em vários tamanhos.

A pasta “p4a” é a principal do framework. Nela estão as bibliotecas, os objetos, os locales e mais alguns componentes importantes.

Os temas devem ficar na pasta “themes”

Exemplo de Estrutura Mínima de uma Aplicação em P4A

```
<ola_mundo>
index.php
   <objects>
    ola_mundo.php
    ola_mascara.php
```

Caso a aplicação utilize algum upload deverá ter uma pasta extra “uploads”, contendo uma sub pasta “tmp” e ambas com permissão 777.

Permissões: os demais arquivos: index.php, ola_mundo.php e ola_mascara.php devem ter permissão de leitura, escrita e execução para o dono e leitura e execução para Outros.

Primeiro Exemplo – Olá Mundo

1)Criamos uma pasta ola_mundo dentro da pasta applications.

2)Criamos dentro de ola_mundo o arquivo index.php com o seguinte conteúdo:

index.php

```
<?php
 // Carregar a biblioteca do framework
require_once( dirname(__FILE__) . '/../../p4a.php' );

// Instanciar a classe que define a aplicação ola_mundo.
// A chasse encontra-se no diretório objects em um arquivo
// com o mesmo nome da classe
$app =& p4a::singleton("ola_mundo");

// Lançar a aplicação
$app->main();
?>
```

No arquivo index.php existe uma chamada para a classe principal do aplicativo que encontra-se no arquivo de mesmo nome (ola_mundo.php) dentro da pasta objects.

ola_mundo.php

```
<?php

// A classe principal da aplicação sempre estende a classe p4a do framework
class ola_mundo extends P4A{
        // Construtor da classe
        function ola_mundo ()   {
                // A primeira coisa a fazer, sempre é lançar o construtor da
super classa p4a
                parent::p4a();

                // A seguir o código para executar, tipicamente abre a primeira
máscara (masks)
                $this->openMask("mensagem");
        }
}
?>
```

Veja que a classe principal abre o arquivo de máscara (mensagem.php).

mensagem.php

```
<?php

// Para criar uma máscara, vem criar uma classe que herda da classe P4A_Mask.
class mensagem extends P4A_Mask{
        // Construtor da classe
        function mensagem()     {
                // Para a aplicação a primeira coisa que se lança é o construtor
da classe pai
                parent::P4A_Mask();

                // Eles vem instanciar o widget para posicionar sobre a máscara
usando o método
                // build da máscara (ou melhor, da classe P4A_Objects da qual
tudo deriva

                // O objeto box é um Label de texto não modificável.
                // Uma vez instanciado o texto adota a mesma posição, formatável
em HTML
                // acessável com $this->box

                $this->build("p4a_box","box");
                $this->box->setValue("Olá Mundo");

                // O objeto frame é um tipo de container disponível que
                // colabora para o posicionamento relativo (sem tabela html) dos
widget
                //em questão, caso se acesse o widget criado usando o apontador
retornado com build
                $frm =& $this->build("p4a_frame","frame");
                $frm->anchorCenter($this->box);

                // Finalmente visualizamos o container (e o widget contido)
                // em uma de 3 seções do template (por default são
top,menu,main)
                $this->display("main", $frm);
```

```
                $this->box->addAction("onClick");
                $this->intercept($this->box, "onClick", "change_box");
        }
        function change_box()
        {
                if ($this->box->getValue() == "Olá mundo") {
                        $this->box->setValue("Clique para acessar");
                } else {  // Abaixo adiciona um ponto ao final
                        $this->box->setValue($this->box->getValue() . ".");
                }
        }
}
?>
```

A pasta objects pode ou não conter uma sub pasta "masks". Caso criemos a pasta "masks", ela deverá contr os arquivos de interface da aplicação (masks). Em termos de organização e entendimento da estrutura é adequada a criação da subpasta masks.

Experimente criar este pequeno exemplo com e sem a pasta masks. Nela apenas ficará o arquivo mensagem.php. Dica: instale a extensão "Web develop" no Firefox e sempre que houver algum erro, corrija o erro e clique em Cookies – Clear Session Cookies e só então clique no botão Refresh do browser.

Se no Internet Explorer, feche e abra o browser novamente após qualquer alteração, ou faça alguma configuração correspondento a da extensão do FF.

Segundo Exemplo – Acessando Banco de Dados

Este exemplo usa um banco de nome "p4a", com o SGBD PostgreSQL. Teremos apenas uma tabela, funcionarios, com a estrutura abaixo:

```
CREATE TABLE funcionarios (
    codigo integer NOT NULL,
    nome character varying(40),
    siape character varying(10),
    foto text,
    observacao text,
    primary key (codigo)
);
```

A estrutura do aplicativo será:

```
<funcionarios>
        index.php
        <objects>
                cadastro.php
                funcionarios.php
        <uploads> (Com permissão de escrita - 777>
                <tmp> (Com permissão de escrita - 777>
```

Arquivos:

index.php

```
<?php

// Ampliar retornos de erro do P4A
define("P4A_EXTENDED_ERRORS", 'TRUE');

// Define locales para pt_BR. Muito útil para o usuário este recurso, já que
campos:
// numéricos receberão a máscara 1.234,56 e campos data
// 25/12/2005 além de outras facilidades
define("P4A_LOCALE", 'pt_BR');

// Define dados da conexão ao banco
define("P4A_DSN", 'pgsql://postgres:suasenha@127.0.0.1:5432/p4a');

//define("P4A_DSN", 'mysql://root:@127.0.0.1/p4a');
//Incluir lib P4A
require_once( dirname(__FILE__) . '/../../p4a.php' );

/*
Now we instance the application with the singleton
method, than we call the "main" method. main is executed
every page call (click and reload included).
The application must be a class under the "objects" directory
and have to extend "p4a" class.
Attention, in p4a all object have to be assigned using the "=&"
operator or you will loose all references. You also have to
use the "&" operator in the method definition, when the method
returns objects (also for the class constructor).
Take a look at "cadastro" class for a better understanding.
*/

// Checar instalação e configuração.
// As linhas abaixo devem ser removidas após a primeira execução
/*
$check = p4a_check_configuration();

// Here we go
if (is_string($check)) {
        print $check;
} else {
        $p4a->main();
}
*/
$app =& p4a::singleton("cadastro");
$app->main();
?>
```

objects/cadastro.php

```
<?php
 // A classe da aplicação sempre estende a classe p4a do framework
class cadastro extends P4A{
        // Construtor da classe
        function cadastro()     {
                // A primeira coisa a fazer sempre é lançar o construtor da
classe p4a
                parent::p4a();
```

```
                // Construção do menu
                $this->build("p4a_menu", "menu");
                $this->menu->addItem("cadastro", "C&adastro");

                $this->menu->items->cadastro-
>addItem("funcionarios","Funcionários");
                $this->intercept($this->menu->items->cadastro->items-
>funcionarios,
                                                "onClick", "menuClick");

                //Em ... addItem("funcionarios","Funcionários"),
                // funcionarios é a classe e Funcionários o label.

                $this->menu->items->cadastro-
>addItem("funcionarios2","Funcionários2");
                $this->intercept($this->menu->items->cadastro->items-
>funcionarios2,"onClick",
                                                            "menuClick");

                // Código da execução, tipicamente abre a primeira máscara
(mask)
                $this->openMask("funcionarios");
        }

        function menuClick()    {
                $this->openMask($this->active_object->getName());
        }
}
?>
```

objects/funcionarios.php

```
<?php
// Para criar uma máscara, cria uma classe que herda da classe P4A_Mask
class Funcionarios extends P4A_Mask{
        // Construtor da classe funcionarios
        function Funcionarios() {
                //Para a aplicação a primeira coisa se lança o costrutor da
classe pai
                $this->p4a_mask();
                $p4a =& p4a::singleton();

                // DB Source
                $this->build("p4a_db_source", "source");
                /*$this->source->setFields(array("funcionarios.*" => "*"));*/
                $this->source->setTable("funcionarios");
                $this->source->setPk("codigo");
                $this->source->addOrder("nome");
                $this->source->setPageLimit(10);
                $this->source->load();
                //$this->source->fields->codigo->setSequence("funcionarios");
                $this->source->fields->codigo->setSequence("codigo");

                $this->setSource($this->source);
                $this->source->firstRow();

                // Personalizar propriedades dos campos (vide função abaixo)
                $this->setFieldsProperties();
```

```
    $fields =& $this->fields;

    // Busca em registros
    $fs_search =& $this->build("p4a_fieldset","fs_search");
    $fs_search->setTitle("LocalizaÃ§Ão de FuncionÃ¡rios");
    $txt_search =& $this->build("p4a_field", "txt_search");
    $txt_search->addAction("onReturnPress");
    $this->intercept($txt_search, "onReturnPress","search");
    $txt_search->setLabel("Nome ou parte");
    $cmd_search =& $this->build("p4a_button","cmd_search");
    $cmd_search->setValue("Ok");
    $this->intercept($cmd_search, "onClick","search");
    $fs_search->anchor($txt_search);
    $fs_search->anchorLeft($cmd_search);

    // Toolbar
    $this->build("p4a_standard_toolbar", "toolbar");
    $this->toolbar->setMask($this);

    // Tabela (exibe funcionários, um registro por linha)
    $table =& $this->build("p4a_table", "table");
    $table->setWidth(725);
    $table->setSource($this->source);
    $table->setVisibleCols(array("codigo","nome","siape"));
    $table->cols->codigo->setLabel("CÃ³digo");

    while ($col =& $table->cols->nextItem()) {
            $col->setWidth(150);
    }
    $table->showNavigationBar();

    // Mensagem de erro
    $message =& $this->build("p4a_message", "message");
    $message->setWidth("300");

    //Fieldset con l'elenco dei campi
    $fset=& $this->build("p4a_fieldset", "frame");
    $fset->setTitle("Cadastro do FuncionÃ¡rio");

    $fset->anchor($this->fields->codigo);
    $fset->anchor($this->fields->nome);
    $fset->anchor($this->fields->siape);
    $fset->anchor($this->fields->foto);
    $fset->anchor($this->fields->observacao);

    // Frame (abriga mensagem, tabela, busca e campos)
    $frm=& $this->build("p4a_frame", "frm");
    $frm->setWidth(730);
    $frm->anchor($fs_search);
    $frm->newRow();
    $frm->anchorCenter($message);
    $frm->anchor($table);
    $frm->anchor($fset);

    // Campos requeridos
$this->mf = array("codigo", "nome", "siape");
    foreach($this->mf as $mf){
            $fields->$mf->label->setFontWeight("bold");
    }
```

```
                // Exibir controles (Display)
                $this->display("main", $frm);
                $this->display("menu", $p4a->menu);
                $this->display("top", $this->toolbar);
        }

        function main() {
                parent::main();
                foreach($this->mf as $mf){
                        $this->fields->$mf->unsetStyleProperty("border");
                }
        }
        // Função que personaliza propriedades dos campos
        function setFieldsProperties()  {
                $p4a =& p4a::singleton();
                $fields =& $this->fields;

                $fields->codigo->setLabel("CÃ³digo");
                $fields->codigo->setWidth(200);
                $fields->codigo->enable(false);

                $fields->nome->setWidth(400);

                $fields->siape->setLabel("SIAPE");
                $fields->siape->setWidth("400");

                $fields->foto->setType("image");

                $fields->observacao->setType("rich_textarea");
                $fields->observacao->enableUpload();
        }
        // Função que salva os registros
        function saveRow()      {
                $valid = true;

                foreach($this->mf as $mf){
                        $value = $this->fields->$mf->getNewValue();
                        if(trim($value) === ""){ // Abaixo, em vermelho as
bordas dos campos
                                $this->fields->$mf->setStyleProperty("border",
"1px solid red");
                                $valid = false;
                        }
                }

                if ($valid) {
                        parent::saveRow();
                }else{
                        $this->message->setValue("Favor preencher todos os
campos requeridos!");
                }
        }
        // Função que implementa a busca
        function search()       {
                $value = $this->txt_search->getNewValue();
                $this->data->setWhere("nome ILIKE '%{$value}%'"); // Busca case-
insensitive
                $this->data->firstRow();
```

```
                $num_rows = $this->data->getNumRows();

                if (!$num_rows) {
                        $this->message->setValue("Nenhum nome encontrado
contendo: $value!");
                        $this->data->setWhere(null);
                        $this->data->firstRow();
                }
        }
}
?>
```

Exemplo Com 3 Tabelas

Este é um exemplo que acompanha o P4A original,contendo 3 tabelas, com exemplos de joins e que roda bem no MySQL. No PostgreSQL, devido a algumas incompatibilidades com a PEAR/DB, ele não roda bem. Segundo o autor do P4A, Fabrizi, esta incompatibilidade é apenas com os joins.

A estrutura das tabelas é:

```
CREATE TABLE brands (
  brand_id INTEGER NOT NULL,
  description TEXT NOT NULL,
  visible BOOL NOT NULL,
  PRIMARY KEY(brand_id)
);

CREATE TABLE categories (
  category_id INTEGER NOT NULL,
  description TEXT NOT NULL,
  visible BOOL NOT NULL,
  PRIMARY KEY(category_id)
);

CREATE TABLE products (
  product_id VARCHAR(50) NOT NULL,
  brand_id INTEGER NOT NULL,
  category_id INTEGER NOT NULL,
  model TEXT NOT NULL,
  purchasing_price DECIMAL(10,2) NOT NULL,
  selling_price DECIMAL(10,2) NOT NULL,
  discount INTEGER NOT NULL,
  little_photo TEXT NULL,
  big_photo TEXT NULL,
  is_new BOOL NOT NULL,
  visible BOOL NOT NULL,
  description TEXT NOT NULL,
  PRIMARY KEY(product_id)
);
```

Crie o banco p4a no MySQL com as tabelas acima.

A estrutura dos arquivos da aplicação:

```
<products_catalogue>
```

```
index.php
<objects>
        products_catalogue.php
        products.php
        categories.php
        brands.php
<uploads> (Com permissão de escrita - 777>
        <tmp> (Com permissão de escrita - 777>
```

Observe que:

index.php em $p4a =& p4a::singleton("products_catalogue") chama a classe products_catalog no arquivo de mesmonome.

products_catalog.php em $this->openMask("products") chama a classe products no arquivo products.php

Através do menu se tem acesso as demais classes dos arquivos brands.php e categories.php.

O código do aplicativo acompanha o P4A, portanto não há necessidade de colar aqui, somente colarei alguns trechos para comentar:

objects/products_catalogue.php

Neste arquivo encontramos este trecho de código:

```
                // Data sources
                $this->build("p4a_db_source", "brands");
                $this->brands->setTable("brands");
                $this->brands->setPk("brand_id");
                $this->brands->addOrder("description");
                $this->brands->load();
                $this->brands->fields->brand_id->setSequence("brands");

                $this->build("p4a_db_source", "categories");
                $this->categories->setTable("categories");
                $this->categories->setPk("category_id");
                $this->categories->addOrder("description");
                $this->categories->load();
                $this->categories->fields->category_id-
>setSequence("categories");
```

Este diz quais as duas outras tabelas (db_sources) que serão utilizadas no aplicativo.

Trecho de “objects/products.php” definindo uma combo com acesso ao banco:

```
                $fields->brand_id->setLabel("Brand");
                $fields->brand_id->setWidth(200);
                $fields->brand_id->setType("select");
                $fields->brand_id->setSource($p4a->brands);
                $fields->brand_id->setSourceDescriptionField("description");
```

O index.php deve ficar parecido com este. Ajuste para os dados do seu MySQL:

```
<?php
// Caso apareça algum erro, é útil descomentar alinha abaixo para melhor debug
```

```
//define("P4A_EXTENDED_ERRORS", 'TRUE');
define("P4A_LOCALE", 'en_US');
define("P4A_DSN", 'mysql://root:@localhost/p4a');
//define("P4A_DSN", 'pgsql://postgres:postabir@localhost:5432/p4a');

require_once dirname(__FILE__) . '/../../p4a.php';

// Check Installation and configuration.
// This lines should be removed after the first run.
$p4a =& p4a::singleton("products_catalogue");
$check = p4a_check_configuration();

// Here we go
if (is_string($check)) {
        print $check;
} else {
        $p4a->main();
}
?>
```

Dicas Extras

Suporte ao Ajax

A versão 1.99.2 já traz suporte ao Ajax. Basta ver o calendário adicionado ao lado dos campos data.

Acentuação

Quando à acentuação vale alertar que a codificação do P4A é a UTF8. Caso tenha algum problema com acentuação deverá encontrar um editor com suporte a essa codificação para corrigir os acentos, como é o caso dos editores do Linux gEdit e Kate.

Dicas do forum do projeto (no sourceForge.net).

Tenho duas tabelas. Quando clicar no botão Save, na Toolbar, quero salvar as duas tabelas.

R - Sobrescreva o método saveRow() na sua mask. Se você tem 2 db_sources você pode fazer:

parent::saveRow(); $outro_db_source->saveRow();

E assim por diante.

- Eu tenho duas mask e preciso compartilhar dados entre ambas. Ao clicar no registro eu preciso abrir a outra mask

R – Você pode acessar dados de qualquer mask com: $p4a->masks->nomemask->desejado

Como interceptar o método Save na Toolbar?

R - $this->intercept($toolbar->button->save, “onClick”, “method”)

Como inserir dados?

R- Você podecriar um db_source e fazer e circular por todos os elementos que você deseja inserir

```
$db_source->newRow();
$db_source->setValue("valor");
$db_source->saveRow();
```

Como alterar Labels?

R -

$this->fields->fieldname->setLabel("Meulabel");

$table->cols->nomecoluna->setLabel("Meulabel");

Como exibir em um campo data a data de hoje por default?

R-

$source->fields->order_date->setDefaultValue(P4A_Date::now(P4A_DATE));

Ou

$format = '%d/%m/%Y-%H/%M/%S'; // Ou outro formato

.

.

P4A_Date::now($format);

Instalando Novos Temas no P4A

Acesse a página - http://p4a.sourceforge.net/icons-packs

E faça o download do tema desejado. Veja que para instalar basta editar a index.php do P4A e adicionar a linha:

define("P4A_ICONS_PATH", "/path/to/the/icons/directory");

Atualmente (06/2007) o P4A está na versão estável 2.0.6 e 2.1.3 em desenvolvimento.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/P4A"
Page categories: PHP

Aplicativos em PHP/FrameWorks/Zend

De Wikibooks

< Aplicativos em PHP | FrameWorks

O Zend Framework é o framework desenvolvido por desenvolvedores da engine do PHP.

Site oficial - http://framework.zend.com

Zend Framework para um PHP mais poderoso

http://felipetonello.com/blog/2007/02/28/zend-framework-para-um-php-mais-poderoso/

Getting Started with the Zend Framework

http://akrabat.com/wp-content/uploads/getting-started-with-the-zend-framework_123.pdf

Iniciando com Zend Framework

O mesmo tutorial anterior traduzido

http://akrabat.com/wp-content/uploads/iniciando-com-zend-framework_130.pdf

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/Zend"
Page categories: PHP

Aplicativos em PHP/FrameWorks/Seagull

De Wikibooks

< Aplicativos em PHP | FrameWorks

Framework Seagull

Seagull é um framework OOP maduro para a construção de aplicativos web, aplicações em linha de comando e em GUI. Licenciado sob BSD, o projeto permite que desenvolvedores PHP facilmente integrem e gerenciem seus códigos e construam aplicações complexas rapidamente.

Site oficial - http://seagull.phpkitchen.com/

Download - http://sourceforge.net/project/showfiles.php?group_id=92482

Instalação

- O diretório seagull/var deve ter permissão de escrita e também o seagull/www/images.

- Chame no navegador http://localhost/seagull/www

- Siga os passos de uma instalação interativa

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/Seagull"
Page categories: PHP

Aplicativos em PHP/FrameWorks/CodeIgniter

De Wikibooks

< Aplicativos em PHP | FrameWorks

Framework CodeIgniter

O Code Igniter é um framework de desenvolvimento de aplicações - algo como um kit de ferramentas - para quem constrói sites em PHP.

Seu objetivo, através de um abrangente conjunto de bibliotecas voltadas as tarefas mais comuns, de uma interface e uma estrutura lógica simples para acesso àquelas bibliotecas, é possibilitar que você desenvolva projetos mais rapidamente do que se estivesse codificando do zero.

Code Igniter permite que você mantenha o foco em seu projeto minimizando a quantidade de código necessário para uma dada tarefa.

Site Oficial - http://www.codeigniter.com/

Site com vários tutoriais e exemplos - http://swik.net/codeigniter+Tutorial

Download - http://codeigniter.com/download.php

Tutoriais em Vídeo - http://codeigniter.com/tutorials/

User Guide - http://codeigniter.com/user_guide/

User Guide(Brasileiro) - http://www.plasmadesign.com.br/codeigniter/user_guide-pt_BR/

Forum - http://codeigniter.com/forums/

Construídos com CodeIgniter - http://codeigniter.com/projects/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/CodeIgniter"
Page categories: PHP

Aplicativos em PHP/FrameWorks/QCodo

De Wikibooks

< Aplicativos em PHP | FrameWorks

9.5 - Framework QCodo

Ótimo Framework que também gera aplicativos completos apenas com a indicação do banco.

Site oficial - http://www.qcodo.com/

Documentação - http://www.qcodo.com/documentation/

1 - Download - http://www.qcodo.com/downloads/

Faça o download do official release para seu distema e descompacte. Descompacte o arquivo. Crie uma pasta "qcodo" no seu diretório web e copie todo o conteúdo da pasta wwwroot para esta pasta qcodo criada.

Edite o arquivo includes/_configuration.php

2 - Defina o diretório do qcodo

define ('DOCROOT_SUBFOLDER', '/qcodo');

Caso esteja usando o qcodo em um diretório virtual ajuste adequadamente

DOCROOT_VIRTUAL_DIRECTORY no arquivo includes/_configuration.php

3 - Defina os dados do SGBD

define('DBCONFIG_1', 'Adapter=MySqli5Database; Server=localhost; Port=; Database=test; Username=root; Password=; Profiling=false; ScriptPath=;');

Faça as devidas adaptações: Server, Port, Database, Username, Password. Por exemplo:

define('DBCONFIG_1', 'Adapter=MySqli5Database; Server=localhost; Port=; Database=condominio; Username=root; Password=; Profiling=false; ScriptPath=;');

Atualmente suporta:

MySqlDatabase (MySQL v4.x, using the old mysql extension)

MySqliDatabase (MySQL v4.x, using the new mysqli extension)

MySqli5Database (MySQL v5.x, using the new mysqli extension)

SqlServerDatabase (Microsoft SQL Server)

PostgreSqlDatabase (PostgreSQL)

4 - Abra o browser em http://localhost/qcodo/_devtools/codegen.php para gerar o aplicativo para o banco indicado

5 - Abra o browser em http://localhost/qcodo/form_drafts/ para ver sua aplicação gerada.

Alerta: caso mude o arquivo de configuração para trabalhar com outro banco, antes renomeie o diretório forms_draft para o nome da aplicacao e mova para o raiz web (sugestão).

Depois recrie o diretório forms_draft e copie para ele apenas o index.php que existia.

Antes de rodar a aplicação gerada em outro diretório:

- altere o include_path do /opt/lampp/etc/php.ini adicionando o caminho do diretório includes, como por exemplo:

include_path = ".:/opt/lampp/lib/php:/opt/lampp/htdocs/qcodo/includes"

- Após copiar o aplicativo para outro diretório e adicionar o includes no include_path altere a terceira linha do

```
index.php para que fique assim:

require('prepend.inc');
```

Este pequeno tutorial visa apenas a geração de código com o qcodo e para isso devemos ter permissão de escrita em vários diretórios. A maneira segura, em sistemas Unix/Linux, sugerida no _README.txt é rodar:

chmod -R ugo+w diretório qcodo.

No Windows, clique com o botão direito no diretório qcodo e Propriedades - Geral - Desmarcar

Somente leitura - Aplicar as alterações a esta pasta, subpastas e arquivos.

Mais detalhes, veja a documentação no site e no _README.txt que acompanha o framework.

A equipe também oferece hospedagem para sites que usem o qcodo:

http://www.qcodohosting.org/page_start.php

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/QCodo"
Page categories: PHP

Aplicativos em PHP/FrameWorks/Zephyr-php

De Wikibooks

< Aplicativos em PHP | FrameWorks

Guia Rápido de Instalação e Uso do Framework ZEPHYR

Site oficial - http://zephyr-php.sf.net

Conteúdo

- 1 Introdução
- 2 Características
- 3 Instalação
- 4 Iniciando a Usar o Zephyr
- 5 Conexão com o banco de dados
- 6 Operações com Bancos de Dados
- 7 Créditos

Introdução

Um Framework baseado em AJAX e MVC para desenvolvedores de PHP5

Podemos desenvolver aplicações comerciais facilmente usando um framework robusto.

Ele é extremamente fácil de aprender e muito simples de implementar.

Com ele podemos entregar uma aplicação comercial completa em AJAX e em camadas no backend (MVC) dentro de alguns minutos.

Instalação e desenvolvimento de pacotes que você desenvolve para Zephyr são agradáveis.

Alé disso devemos receber todas os recursos do mais popular template engine "smarty" e da poderosa camada de abstração de bancos de dados "ADOdb".

Vamos seguir com a documentação para descobrir como é fácil desenvolver aplicações AJAX usando o framework Zephyr.

Zephyr é um framework que utiliza MVC (model-view-controller) e suporta completamente a separação da camada de negócio da camada de apresentação. O framework foi construído usando bibliotecas populares testadas no backend.

Por isso o código desenvolvido com Zephyr é mais estável e flexível para usuários finais.

Características

Aqui estão algumas das mais importantes características do Zephyr:

1. Usa o template engine Smarty. Assim o desenvolvimento com os templates no zephyr é flexível e estruturado.

```
  Além disso podemos usar todas os poderosos recursos do smarty em nossos
templates.

  Assim a curva de aprendizado é baixa.
```

2. A adoDB é usada como DAL (data access layer) camada de abstração de bancos de dados no zephyr.

3. Podemos gerenciar múltiplos bancos de dados concorrentemente nas ações do zephyr.

4. Fácil integração de scripts personalizados em PHP e de bibliotecas em javascript.

5. O frontend do zephyr é desenvolvido em cima da famosa biblioteca prototype do Sam Stephenson.

```
  Isto reduz o trabalho de usar javascript nos templates e dá a você enorme
flexibilidade.
```

6. Zephyr usa CPAINT como interface para AJAX. Assim você pode facilmente passar múltiplos caracteres de e para suas ações usando zephyr.

7. Cada pacote (package) ou aplicação tem seu próprio inicializador.

8. Implementa processadores de pré ações que ajudam-nos a executar algumas ações específicas antes das ações normais.

```
Isto deve nos ajudar a implementar validações mais robustas.
```

9. Suporta ações do cron que executam em intervalos regulares.

10. Fácil manipulação de registros do banco de dados. Requisições internas do gerente do zephyr gerencia todos os parâmetros requisitados de uma forma flexível e você não precisa acessar e configurar esses parâmetros exclusivamente.

11. Podemos integrar facilmente com qualquer pacote PEAR.

12. Você pode gerenciar funções de agregação de bancos de dados através do objeto interno do zephyr de agregação de bancos de dados.

13. Suporta múltiplos níveos de filtros definidos pelo usuário.

14. Fácil integração com os ezComponents.

Instalação

Download - http://sourceforge.net/project/showfiles.php?group_id=151993

Apenas faça o download e descompacte no diretório web.

Altere as permissões do diretório "temp" para 777.

Uma estrutura típica do zephyr:

```
- zephyr
  | + abstract
  | + documentation
  | + functions
  | + helper
  | + images
  | + interfaces
  | + internal
  | + javascript
  | + packages
  | + temp
  | + thirdparty
  |   .htaccess
  |   index.php
```

Iniciando a Usar o Zephyr

Todas as aplicações no Zephyr estão no diretório "packages".

Caso pretenda criar uma aplicação com o Zephyr, apenas crie um sub-diretório no diretório "packages" ou então faça uma cópia de um dos aplicativos de exemplo. Esta última opção é mais indicada para quem está dando seus primeiros passos.

Basicamente:

Criar o banco - agenda (MySQL)

```
CREATE TABLE `agd` (
  `codigo` int(11) NOT NULL default '0',
  `nome` varchar(255) NOT NULL default '',
  `email` varchar(50) NOT NULL default '',
  `site` varchar(50) NOT NULL default '',
  PRIMARY KEY  (`codigo`)
)
```

Alterar o script dbinfo.class.php no diretório "helper" da aplicação com as informações corretas.

- Faça uma cópia do aplicativo "student_db" para o nome desejado, "agenda" por exemplo

- Então, tendo em vista sua tabela e campos, edite cada um dos arquivos e troque as ocorrencias de student, students, std, por agenda, agendas, agd e os campos também pelos respectivos.

Para facilitar crie a tabela com a mesma quantidade de campos e com tipos idêntivos aos originais.

- Comece editando o arquivo XML (package.xml) que se encontra no raiz da aplicação

- O arquivo readme.txt ou leiame-txt contém algumas informações para rodar a aplicação, inclusive o esquema do banco.

- Depois percorra cada um dos arquivos do aplicativo.

- Para facilitar use o recurso procurar e sobrescrever, tendo o cuidado de ativar a opção procurar palavra inteira.

Aproveite para observar a estrutura das pastas, arquivos, classes e funções do framework.

Veja a estrutura:

```
- zephyr
  | - packages
  |    | + student_db
  |    | + agenda
```

Script para o PostgreSQL:

```
CREATE TABLE agd (
  codigo int NOT NULL default 0,
  nome varchar(255) NOT NULL default '',
  email varchar(50) NOT NULL default '',
  site varchar(50) NOT NULL default '',
  PRIMARY KEY  (codigo)
)
```

Para suporte ao postgresql adicione este código:

```
case "postgres":
     $dsn =  "{$dbinfo->get_dbtype()}://{$dbinfo->get_dbuser()}:{$dbinfo-
>get_dbpwd()}@{$dbinfo->get_dbhost()}/{$dbinfo->get_dbname()}?persist={$dbinfo-
>get_persist()}";
     break;
```

Logo após a entrada do mysql no arquivo zephyr/helper/DatabaseConector.class.php

Veja aqui um exemplo que criei partindo do student_db (agenda, 7,98Kb).

Conexão com o banco de dados

Os dados encontram-se na classe dbinfo.class.php dentro do diretório helper:

```
<?
class dbinfo extends abstractdbinfo {
    public function __construct()
    {
            $this->dbhost = "localhost";
        $this->dbname = "agenda";
        $this->dbuser = "root";
        $this->dbpwd = "";
        $this->dbtype = "mysql";
        $this->persist = 1;
    }
}
?>
```

Para PostgreSQL

```
<?
class dbinfo extends abstractdbinfo {
    public function __construct()
    {
        $this->dbhost = "localhost";
        $this->dbname = "agenda";
        $this->dbuser = "postgres";
        $this->dbpwd = "postgres";
        $this->dbtype = "postgres";
        $this->persist = 1;
    }
}
?>
```

O Zephyr vem uma estrutura mínima do ADOdb, que atualmente traz:

- MySQL

- SQLite

- Postgresql

- MSSQL (Both ODBC and Normal)

- ODBC

- Oracle

Mas caso queira trabalhar com qualquer outro SGBD nao contemplado, apenas baixe o ADODB e utilize sem problema.

Apenas para o SQLite, que é baseado em arquivos, existe algumas diferenças. Veja:

Devemos criar um diretório para o banco "sqlitedb".

```
<?
class dbinfo extends abstractdbinfo
{
    public function __construct()
    {
        $pm = new packagemanager();
```

```
        $package_path = $pm->get_package_path(); //physical path to this package
        $this->dbhost = $package_path."/sqlitedb/";
        $this->dbname = "test.sqlite";
        $this->dbtype = "sqlite";
        $this->persist = 1;
    }
}
?>
```

Operações com o SQLite:

//viewreport.class.php

```
<?
class viewreport implements action
{
    public $params;
    public function execute()
    {
        $dao = new DAO(); //instantiate Data Access Object
        $dao->execute("create table students (name varchar(100))");

        $dao->execute('insert into students values("ayesha")');
        $dao->execute('insert into students values("didar")');
        $dao->execute('insert into students values("shahana")');
        $dao->execute('insert into students values("tipoo")');

        $result = $dao->selectBySQL("select * from students");
        $data = array("students"=>$result);
        return array("view_file"=>"reports","data"=>$data);
    }
}
?>
```

Veja agora na camada view:

```
//reports.tpl
<h1>All Students</h1>
{foreach item=student from=$students name=std}
{$smarty.foreach.std.iteration}.{$student.name}<br>
{/foreach}
```

Operações com Bancos de Dados

Inserir registos em tabelas com o framework Zephyr é muito fácil:

Veja a estrutura da tabela student:

```
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| name         | varchar(255)|      |     |         |                |
| roll         | int(11)     |      |     |         |                |
| class        | int         |      |     |         |                |
+--------------+-------------+------+-----+---------+----------------+
```

Agora vamos criar o modelo de domínio, que é uma classe simples, contendo todos os campos, exceto os de tipo auto_increment. Então para a classe student:

```
class student
{
    public $name;
    public $roll;
    public $class;
}
```

Agora vamos criar um arquivo na camada view para entrada dos dados:

```
     <input type="text" id="name">
     <input type="text" id="roll">
     <input type="text" id="class">
     <input type="button" value=" = " onclick="insert_std();" />
```

Veja a função:

```
function insert_std()
{
  serialized = group_serialize("name","roll","class");
  load_action_value('insert_record', serialized,'result');
}
```

Esta função serializa os dados e invoca a ação "insert_record". Veja a ação:

```
<?
load_db_domain("student");

class insertstudent implements action {

    public $params ;

    public function execute()
    {
        $cls =     auto_fill_domain("student");
        $dao = new DAO($cls);
        $query = $dao->insert();
        $students = $dao->selectBySQL("SELECT * FROM   students");
        $data = array("students"=>$students);

        return array("view_file"=>"liststudents", "data"=>$data);
    }

}
?>
```

Alterando e excluindo registros:

```
<?
load_db_domain("student");

class updatestudent implements action {

    public $params ;

    public function execute()
    {
```

```
        cls =     auto_fill_domain("student");
        $dao = new DAO($cls);
        $query = $dao->update("roll=1");
        $students = $dao->selectBySQL("SELECT * FROM   students");
        $data = array("students"=>$students);

        return array("view_file"=>"liststudents", "data"=>$data);
    }
}
?>
```

Faz a tualização "where roll = 1".

Créditos

Lead developer : Hasin Hayder

Javascript and Frontend : Rasiul Kabir

DAO and Helper classes : Prashun Paul

Feedback - Equipe de Desenvolvimento (zephyr@phpxperts.net or zephyr@phpxperts.org)

Forum - http://zephyr-forum.phpxperts.net

Documentação online e offline distribuida juntamente com o pacote para download.

Para maiores detalhes (em inglês), veja a documentação oficial.

A intenção deste pequeno guia é apenas a de facilitar a vida de quem não tem um bom domínio do inglês.

Tradução direto do tutorial original do site por Ribamar FS - ribafs[ ] yahoo.com - 22:06 28/11/2006

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/Zephyr-php"
Page categories: PHP

Aplicativos em PHP/FrameWorks/CakePHP

De Wikibooks

< Aplicativos em PHP | FrameWorks

Conteúdo

- 1 Framework CakePHP
- 2 Muitos Tutoriais sobre CakePHP
- 3 Scaffolding with CakePHP - Managing Your Fantasy Football Team
- 4 CakePHP tutorial no. 2 from IBM
- 5 Tutorial da IBM em 5 partes (Cook up Web sites fast with CakePHP)
- 6 Joining powers of two great systems: Joomla and CakePHP
- 7 Finally a practical solution: Joomla with CakePHP together - JAKE
- 8 Joomla-CakePHP bridge
    - 8.1 Configurações para o Joomla-CakePHP bridge
- 9 Scaffolding a bookmarks manager
- 10 Vídeo Aula WebBaker do Cake

Framework CakePHP

Um framework em PHP que se inspirou no famoso Ruby on Rails.

Site Oficial - http://cakephp.org/

Downloads - http://cakephp.org/downloads

Muitos Tutoriais sobre CakePHP

http://swik.net/cakephp?page=1

Tutorial do Elton - Desenvolva rapidamente utilizando o framework CakePHP - http://www.eltonminetto.net/docs/cake_fisl.pdf

Tutorial do Carlos Silva na Revista PHP - http://www.revistaphp.com.br/artigo.php?id=93

CakePHP Brasil - http://www.cakephp.com.br/

Tutorial - http://www.cakephp.com.br/modules/news/index.php?storytopic=0&storynum=5&start=10

Scaffolding with CakePHP - Managing Your Fantasy Football Team

http://www.developer.com/lang/php/print.php/3636686

CakePHP tutorial no. 2 from IBM

http://www.scribd.com/doc/5545/CakePHP-tutorial-no-2-from-IBM

Tutorial da IBM em 5 partes (Cook up Web sites fast with CakePHP)

http://www.ibm.com/developerworks/views/opensource/libraryview.jsp?search_by=cook+web+sites+CakePHP

Cadastre-se gratuitamente para ter acesso ao tutorial (login é o e-mail).

Joining powers of two great systems: Joomla and CakePHP

http://www.gigapromoters.com/blog/2007/01/28/joining-powers-of-two-great-systems-joomla-and-cakephp/

Finally a practical solution: Joomla with CakePHP together - JAKE

http://www.gigapromoters.com/blog/2007/02/13/finally-a-practical-solution-joomla-with-cakephp-together-jake/

Joomla-CakePHP bridge

É um componente para Joomla que permite executar aplicações Cake no CMS Joomla sem qualquer alteração do código.

http://dev.sypad.com/projects/jake/

Configurações para o Joomla-CakePHP bridge

http://dev.sypad.com/projects/jake/documentation/configuration/

Scaffolding a bookmarks manager

scaffold é o gerador de aplicativos do Cake, que usa o script bake.php para isso.

http://grahambird.co.uk/cake/tutorials/scaffolding.php

Vídeo Aula WebBaker do Cake

http://cakephp.com.br/modules/mydownloads/visit.php?lid=8

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/CakePHP"
Page categories: PHP

Aplicativos em PHP/FrameWorks/Symfony

De Wikibooks

< Aplicativos em PHP | FrameWorks

Predefinição:Info/Software

Symfony é um framework web escrito em PHP que segue o paradigma MVC (model-view-controller). Disponível sob a Licença MIT, Symfony é um Software livre.

Os criadores do **Symfony** afirmam que a curva de aprendizagem necessária para seu uso proficiente está reduzida a menos de um dia. Ele é projetado para permitir que os desenvolvedores apliquem princípios ágeis do desenvolvimento (tais como DRY, KISS ou XP) e foquem na regras de negócio sem necessitar escrever muitos arquivos de configuração XML, comuns nos frameworks atuais.

Symfony ainda tem como objetivo construir aplicações robustas em contexto empresarial, e dar aos desenvolvedores controle total sobre a configuração: da estrutura de diretórios às bibliotecas externas, quase tudo pode ser personalizado. Para atingir os guidelines de desenvolvimento das empresas, o framework é empacotado com ferramentas adicionais para ajudar os desenvolvedores nos testes, depuração e documentação dos projetos.

Symfony é usado pelo serviço de perguntas e respostas Askeet e por muito mais aplicações, incluindo o Yahoo Bookmarks, aplicação com mais de 20 milhões de usuários.

Conteúdo

- 1 Referências
- 2 Leitura recomendada
- 3 Ligações externas
- 4 24 Tutoriais do Symfony
- 5 Learn symfony: a Beginner's Tutorial
- 6 How to setup a symfony project in Trustudio or PHPEclipse
- 7 Meu primeiro projeto com Symfony

Referências

Leitura recomendada

- Potencier, Fabien and Zaninotto, François. (2007). *The Definitive Guide to Symfony.* Apress. ISBN 1590597869.

Ligações externas

- Symfony Project - Site oficial do projeto.

- Symfony Brasil - Site da comunidade brasileira.
- symfony-pt - Lista de discussão da comunidade brasileira.

24 Tutoriais do Symfony

Ótimos tutoriais. O do dia 1 está traduzido também para português.

http://www.symfony-project.com/askeet

Link para o PDF de todos os tutoriais - http://www.symfony-project.com/downloads/askeet/askeet.pdf

O Download do Projeto resultante dos 24 tutoriais - http://www.symfony-project.com/downloads/askeet_release_1_0.tgz

Learn symfony: a Beginner's Tutorial

http://www.sitepoint.com/print/symfony-beginners-tutorial

Código fonte do tutorial - http://www.sitepoint.com/examples/symfony/sandbox_final.tgz

How to setup a symfony project in Trustudio or PHPEclipse

http://trac.symfony-project.com/trac/wiki/SymfonyWithEclipseBasedEditors

Meu primeiro projeto com Symfony

Resumo do tutorial "My first symfony project" do site oficial.

- Certifique-se de que está usando PHP5

- Ative o suporte a SQLite no php.ini, descomentando a "extensão extension=php_sqlite.dll"

- Fazer o download do sf_sandbox de http://www.symfony-project.com/get/sf_sandbox.tgz e descompactar no diretório web

- Abra no navegador http://localhost/sf_sandbox/web/index.php para testar.

- Inicializar o modelo de dados:

Edite o script sf_sandbox/config/schema.yml e cole as linhas abaixo

propel:

```
 weblog_post:
   _attributes: { phpName: Post }
   id:
   title:       varchar(255)
   excerpt:     longvarchar
   body:        longvarchar
   created_at:
 weblog_comment:
   _attributes: { phpName: Comment }
   id:
   post_id:
   author:      varchar(255)
   email:       varchar(255)
   body:        longvarchar
   created_at:
```

Caso queira traduzir, fique à vontade com exceção de created_at.

Este script define a estrutura de duas tabelas que serão criadas com o SGBD SQLite.

Obs.: Para facilitar adicione o diretório do seu PHP ao path.

Obs2.: Se usando o Xampp 1.6.2 e receber mensagens de erro ao executar o comando php, então comente a linha com a extensão ";extension=php_svn.dll" no arquivo xampp\php\php.ini.

Salve o arquivo e abra o prompt de comando no diretório sf_sandbox e execute:

php symfony propel-build-model

Esteja certo de estar no raiz do diretório "sf_sandbox". Este comando criará algumas classes no diretório sf_sandbox/lib/model. Estas classes permitem acesso ao banco sem se digitar uma linha de código SQL. Symfony usará a biblioteca Propel para isso. Chamaremos estes objetos de model.

Agora entre o seguinte comando:

php symfony propel-build-sql

Um arquivo lib.model.schema.sql será criado em sf_sandbox/data/sql/. Podemos criar um banco em outro SGBD mas por default o projeto sf_sandbox deve usar o arquivo sandbox.db em sf_sandbox/data.

Para criar a estrutura das tabelas baseadas no arquivo SQL, entre:

php symfony propel-insert-sql

Criando a Aplicação

O symfony agora criará uma estrutura básica de um CRUD da aplicação que poderá ser sustomizado.

php symfony propel-generate-crud frontend post Post php symfony propel-generate-crud frontend comment Comment php symfony clear-cache

Agora temos dois módulos (pos e comment) que permitirão que você manipule os objetos das classes Post e Comment. Um módulo geralmente representa uma página ou um grupo de páginas com uso similar. Seus novos módulos estão localizados em sf_sandbox/apps/frontend/modules/ e podem ser acessados através das URLs:

http://localhost/sf_sandbox/web/frontend_dev.php/post

http://localhost/sf_sandbox/web/frontend_dev.php/comment

Veja que não digitamos nenhuma linha de código PHP, nem SQL nem sequer trabalhamos com SGBD.

Claro que aqui tem somente o esqueleto mas mesmo assim a ferramenta já nos deu ganho de produtividade.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/Symfony"
Page categories: PHP

Aplicativos em PHP/FrameWorks/Prado

De Wikibooks

< Aplicativos em PHP | FrameWorks

O que é o Prado?

PRADO é um framework de programação baseado em componentes e orientado a eventos para desenvolvimento de aplicações em PHP 5.

Site oficial - http://www.pradosoft.com/

Download - http://www.pradosoft.com/download/

Documentação - http://www.pradosoft.com/documentation/

Alguns Recursos do Prado

Podemos testar online no próprio sites do Prado, antes de instalar, alguns dos seus recursos:

http://www.pradosoft.com/demos/

- Exemplo de blog - Exemplo de chat - Exemplo com Flex - E alguns outros

Para as pessoas que estão começando em framework e querem um bom exemplo de como instalar e usar o Prado fazendo seu primeiro "Olá, mundo!!"

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1886&pagina=1

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/Prado"
Page categories: PHP

Aplicativos em PHP/FrameWorks/ZFRails

De Wikibooks

< Aplicativos em PHP | FrameWorks

ZFRails (Zend Framework on Rails)

O ZFRails (Zend Frameworks on Rails) é uma iniciativa livre com o objetivo de tornar o Framework de Desenvolvimento PHP Zend Framework ainda mais fácil de se trabalhar. O objetivo do projeto é desenvolver uma plataforma semelhante ao Ruby On Rails, mas para a linguagem PHP. Na versão 0.0.1 o software conta com um script com a função de dispatcher e já possui scripts para criação e remoção de controllers e models. Para as próximas versões os seguintes recursos estão em planejamento:

- Criação de Scaffolds

- Melhor Integração com Ajax (criação de classes especiais para Ajax)

- Criação de Novos Plugins/Classes

Se alguém se interessar em colaborar com o projeto entrar em contato pelo email marcelo@diotto.netEste endereço de e-mail está protegido contra spam bots, pelo que o Javascript terá de estar activado para poder visualizar o endereço de email

Download do ZFRails 0.0.1: http://zfrails.diotto.net/

Este texto acima foi colado do site da divulgação da notícia - http://www.diotto.net/site/downloads/programa-o/zfrails-0.0.1/details.html

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/FrameWorks/ZFRails"

Aplicativos em PHP/Abstrações de Bancos de Dados/AdoDB

De Wikibooks

< Aplicativos em PHP | Abstrações de Bancos de Dados

Conteúdo

- 1 Introdução
- 2 Exemplo prático para mostrar alguns recursos da ADOdb
- 3 Alguns aliases para a conexão
- 4 Exemplo com as funções RecordCount() e FieldCount()
- 5 Restringir o número de registros recebidos podemos usar o método SelectLimit()
- 6 Obter uma lista dos bancos e tabelas do SGBD
- 7 Quando precisamos executar uma consulta várias vezes
- 8 Uso de transações
- 9 Cache de Consultas
- 10 Exportar um resultset para diversos formatos
- 11 Debugando
- 12 Recordset Filters
- 13 Inserts and Updates
- 14 MetaTypes
- 15 Apresentando a adobDB

Introdução

ADOdb (Active Data Objects DataBase) é uma abstração de bancos de dados para PHP. Também oferece uma versão para Python.

Atualmente suporta diversos SGBDs: MySQL, Oracle, Microsoft SQL Server, Sybase, Sybase SQL Anywhere, Informix, PostgreSQL, FrontBase, Interbase (Firebird and Borland variants), Foxpro, Access, ADO e ODBC.

Via ODBC podemos também conectar com diversos SGBDs: Progress, SQLite, DB2 e outros.

É uma das mais ricas e eficientes abstrações existentes para PHP atualmente.

Caso não se utilize uma abstração no código, ao migrar para outro SGBD o código sofrerá diversas alterações e além disso exigirá o conhecimento das funções de ambos os SGBDs, mas utilizando algo como ADOdb a alteração será apenas nos dados da conexão sem necessidade de conhecer as funções do PHP com os SGBDs.

Exemplo prático para mostrar alguns recursos da ADOdb

Site oficial - http://php.weblogs.com/ADODB

Download - http://adodb.sourceforge.net/#download

Documentações - http://adodb.sourceforge.net/#docs

Este tutorial teve como base os tutoriais do site oficial e também o ótimo tutorial:

PHP Application Development With ADODB

http://www.devshed.com/c/a/PHP/PHP-Application-Development-With-ADODB-part-1/

Banco de Dados para o exemplo: dbbiblioteca

create database dbbiblioteca (com o postgresql mas para usar outro SGBD basta trocar os dados)

Tabela - biblioteca

```
CREATE TABLE biblioteca (
    id serial NOT NULL PRIMARY KEY,
    titulo character(100),
    autor character(45),
    data date
);
```

Vamos inserir alguns registros:

```
INSERT INTO biblioteca (id, titulo, autor, data) VALUES (1, 'Os Sertões',
'Euclides da Cunha','2005-12-25');
INSERT INTO biblioteca (id, titulo, autor, data) VALUES (2, 'Os Lusíadas',
'Camões','2005-12-25');
INSERT INTO biblioteca (id, titulo, autor, data) VALUES (3, 'A Divina Comédia',
'Dante','2005-12-25');
```

```
INSERT INTO biblioteca (id, titulo, autor, data) VALUES (4, 'Contos',
'Voltaire','2005-12-25');
INSERT INTO biblioteca (id, titulo, autor, data) VALUES (5, 'O Conde de Monte
Cristo', 'Alexandre Dumas Pai','2005-12-25');
```

Fazer o download e descompactar numa pasta do seu DocumentRoot. Neste exemplo descompactei no raiz.

Alguns aliases para a conexão

MySQL - mysql

PostgreSQL - postgres

Interbase/Firebird - ibase

SQLite - sqlite

Vamos agora criar alguns scripts PHP para acessar o banco usando ADOdb.

Esta parte inicial se repetirá em todos os scripts, portanto apenas farei a citação de "inicial" nos demais scripts

inicial

```
<?php
// Descomentar a linha abaixo para visualizar como plaintext no browser
// header("Content-Type: text/plain");

// include the ADODB library
include("adodb/adodb.inc.php");

// create an object instance
// Configurar para uma conexão tipo PostgreSQL
$db = NewADOConnection("postgres"); // MySQL seria "mysql"

// Abrir uma conexão com o banco de dados
// $db->Connect("servidor", "usuario", "senha", "banco")
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Falha
na conexão!");
//final do inicial

<h2>executar a consulta</h2>

$query = "SELECT * FROM biblioteca";
$result = $db->Execute($query) or die("Erro na consulta: $query. " . $db-
>ErrorMsg());

// Iteração através do resultset
// imprimir dados em colunas no formato TÍTULO - AUTOR

while (!$result->EOF){
        echo $result->fields[1] . " - " . $result->fields[2] . "<br>";
        $result->MoveNext(); // Veja que função útil, como também o EOF
                             // (End Of File, enquanto não chegar ao final)
}
```

```
<h2>receber e imprimir o número de registros do resultset com muita
simplicidade</h2>
echo "<br>[" . $result->RecordCount() . " registros retornados]<br>";

<h2>Fechar a conexão com o banco</h2>
$db->Close();

?>
```

ADODB também oferece um número de métodos alternativos para processar um resultset. Por exemplo, você pode receber o resultset como um array associativo indexado de string, onde as chaves são nomes de campos e os valores os correspondentes valores dos campos.

Um exemplo:

inicial - (aqui cole o trecho inicial) referido acima

```
// get resultset as associative array
$ADODB_FETCH_MODE = ADODB_FETCH_ASSOC;

// execute query
$query = "SELECT * FROM biblioteca";
$result = $db->Execute($query) or die("Error in query: $query. " . $db-
>ErrorMsg());

// iterate through resultset
// print column data in format TITLE - AUTHOR
while (!$result->EOF)
{
        echo $result->fields['titulo'] . " - " . $result->fields['autor'] .
"<br>";
        $result->MoveNext();
}

// get and print number of rows in resultset
echo "<br>[" . $result->RecordCount() . " registros retornados]<br>";

// close database connection
$db->Close();

?>
```

Você pode usar o método GetAll() no lugar do Execute() que retorna o resultset completo com um array bidimensional de pares campo-valor. Este array pode então ser processado com um simples "foreach" ou um loop "for".

Um exemplo:

inicial

```
// execute query
$query = "SELECT * FROM biblioteca";
$result = $db->GetAll($query) or die("Error in query: $query. " . $db-
>ErrorMsg());

// clean up
$db->Close();
```

```
// uncomment the following line to see the returned array.
// print_r($result);

// iterate through resultset
// print column data in format TITLE - AUTHOR
foreach ($result as $row)
{
        echo $row[1] . " - " . $row[2] . "<br>";
}

// get and print number of rows in resultset
echo "<br>[" . sizeof($result) . " registros retornados]<br>";

?>
```

Exemplo com as funções RecordCount() e FieldCount()

inicial

```
// execute query
$query = "SELECT * FROM biblioteca";
$result = $db->Execute($query) or die("Error in query: $query. " . $db-
>ErrorMsg());

// get and print number of rows in resultset
echo $result->RecordCount() . " registros retornados<br>";

// get and print number of fields in resultset
echo $result->FieldCount() . " campos retornados<br>";

// clean up
$db->Close();

?>
```

Podemos obter informações sobre cada campo com o método FetchField(), que retorna um objeto contendo informações detalhadas sobre as propriedades dos campos, incluindo seus nomes e tipos. Um exemplo:

inicial

```
// execute query
$query = "SELECT * FROM biblioteca";
$result = $db->Execute($query) or die("Error in query: $query. " . $db-
>ErrorMsg());

// get field information
for($x=0; $x<$result->FieldCount(); $x++) {
        print_r($result->FetchField($x));
}

// clean up
$db->Close();
?>
```

Para a execução de consultas INSERT em tabela contendo chave primária com auto-incremento

podemos obter o último ID gerado do auto-incremento usando o método Insert_ID(). Exemplo:

inicial

```
// execute query
$titulo = $db->qstr("It's Not Me, It's You!");
$autor = $db->qstr("J. Luser");
$query = "INSERT INTO biblioteca (titulo, autor) VALUES ($titulo, $autor)";
$result = $db->Execute($query) or die("Erro na consulta: $query. " . $db-
>ErrorMsg());

// print auto-generated ID
if ($result)
{
   echo "O último ID inserido foi " . $db->Insert_ID();
}

// clean up
$db->Close();
?>
```

Obs.: o método qstr() é usado para "escapar" caracteres especiais em consultas com strings.

Quando estamos utilizando consultas que afetam registros de tabelas, como insert, delete ou update o método Affected_Rows() retorna o número de registros afetados.

inicial

```
// execute query
$query = "DELETE FROM biblioteca WHERE author = 'Euclides da Cunha'";
$result = $db->Execute($query) or die("Erro na consulta: $query. " . $db-
>ErrorMsg());

// return number of affected rows
if ($result)
{
   echo $db->Affected_Rows() . " registros excluídos";
}

// clean up
$db->Close();
?>
```

Restringir o número de registros recebidos podemos usar o método SelectLimit()

inicial

```
// execute query
// receber 2 registros, iniciando do terceiro, ou seja, o terceiro e o quarto
registros
$query = "SELECT * FROM biblioteca";
$result = $db->SelectLimit($query, 2, 3) or die("Erro na consulta: $query. " .
$db->ErrorMsg());

// iterate through resultset
```

```
while (!$result->EOF)
{
        echo $result->fields[1] . " - " . $result->fields[2] . "<br>";
        $result->MoveNext();
}

// clean up
$db->Close();
?>
```

Alerta: Cuidado com o copiar e colar. Este exemplo acima e vários outros acusaram erro ao executar.

Normalmente o erro era na linha (echo $result->fields[1] . " - " . $result->fields[2] . "<br>";).

O original estava no Write do OpenOffice. Normalmente apenas excluo os espaços antes do echo e tá resolvido ou então redigito apenas a linha do erro.

Algumas vezes precisei remover alguns espaços após o ponto e vírgula.

Obter uma lista dos bancos e tabelas do SGBD

Através dos métodos MetaDatabases() e MetaTables().

inicial

```
// get database list
echo "Bancos:<br>";
foreach($db->MetaDatabases() as $d){
        echo "=> $d<br>";
}

// get table list
echo "<br>Tabelas no banco atual:<br>";
foreach($db->MetaTables() as $table)
{
        echo "=> $table<br>";
}

// clean up
$db->Close();
?>
```

Quando precisamos executar uma consulta várias vezes

Como por exemplo diversos INSERTs. O ADOdb conta com diversos recursos úteis. Vejamos:

inicial

```
// prepara a consulta e a deixa em "banho maria", sem a executar ainda
$query = $db->Prepare("INSERT INTO biblioteca (id, titulo, autor) VALUES (?,
 ?, ?)");

// ler lista titulo-autor do arquivo CSV
```

```
$data = file("lista.txt");

// iteração através de cada linha do arquivo
foreach ($data as $l){
        // separa com vírgulas
        $arr = explode(",", $l);
        // insere os valores na consulta preparada anteriormente
        $result = $db->Execute($query, array($arr[0], '$arr[1]', '$arr[2]')) or
die("Erro na consulta: $query. " . $db->ErrorMsg());
}

// clean up
$db->Close;
?>
```

Atentar para o fato de que o prepare apenas deixa a consulta pronta e o execute finaliza a execução.

Isto melhora muito o desempenho quando temos muitas consultas a realizar.

Uso de transações

Caso o SGBD tenha suporte:

inicial

```
// turn off auto-commit
// begin transaction block
$db->BeginTrans();

// first query
$query = "INSERT INTO biblioteca (titulo, autor) VALUES ('Titulo A', 'Autor
B')";
$result = $db->Execute($query) or die("Erro na consulta: $query. " . $db-
>ErrorMsg());

// use ID from first query in second query
if ($result)
{
  $id = $db->Insert_ID();
   $query = "INSERT INTO purchase_info (id, price) VALUES ($id, 'USD 39.99')";
   $result = $db->Execute($query) or die("Error in query: $query. " . $db-
>ErrorMsg());
}

// if no failures
if ($result)
{
   // commit
   $db->CommitTrans();
}
// else rollback
else
{
   $db->RollbackTrans();
}

// clean up
```

```
$db->Close;
?>
```

Cache de Consultas

ADOdb tembém oferece suporte para cache de consultas. Que oferece um excelente ganho de desempenho, principalmente nos casos em que precisamos executar uma mesma consulta diversas vezes.

Para ver a diferença abaixo temos uma consulta normal:

inicial

```
// execute query
$query = "SELECT * FROM biblioteca";
$result = $db->Execute($query) or die("Erroo na consulta: $query. " . $db-
>ErrorMsg());

// iterate through resultset
// print column data in format TITLE - AUTHOR
while (!$result->EOF)
{
        echo $result->fields[1] . " - " . $result->fields[2] . "<br>";
        $result->MoveNext();
}

// get and print number of rows in resultset
echo "<br>[" . $result->RecordCount() . " registros retornados]<br>";

// close database connection
$db->Close();
?>
```

Agora usando o cache de consulta:

inicial

```
// execute query
$query = "SELECT * FROM biblioteca";
$result = $db->CacheExecute(300,$query) or die("Erro na consulta: $query. " .
$db->ErrorMsg());

// iterate through resultset
// print column data in format TITLE - AUTHOR
while (!$result->EOF)
{
        echo $result->fields[1] . " - " . $result->fields[2] . "<br>";
        $result->MoveNext();
}

// get and print number of rows in resultset
echo "<br>[" . $result->RecordCount() . " registros retornados]<br>";

// close database connection
$db->Close();

?>
```

Algo que dá um bom trabalho é criar um select que seja populado de uma tabela. O menu drop-down. O ADOdb tem um método especificamente para esta finalidade.

```
<html>
<head></head>
<body>
<?php
// include the ADODB library
include("adodb/adodb.inc.php");

// create an object instance
// configure it for a PostgreSQL connection
$db = NewADOConnection("postgres");

// open connection to database
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");

// execute query
$query = "SELECT titulo, id FROM biblioteca"; // O primeiro será exibido e o
segundo armazenará o resultado
$result = $db->Execute($query) or die("Erro na consulta: $query. " . $db-
>ErrorMsg());

// print HTML menu
print $result->GetMenu("biblioteca", '', false); // primeiro parâmetro é o nome
do select

// close database connection
$db->Close();
?>
</body>
</html>
```

Exportar um resultset para diversos formatos

```
      - texto separado por vírgula
```

- texto separado por tabulação

- tabela HTML

Estes recursos não fazem parte da classe ADODB, portanto precisaremos importar outras classes.

```
<?php
// uncomment this to see plaintext output in your browser
// header("Content-Type: text/plain");

// include the ADODB library
include("adodb/adodb.inc.php");

// include conversion functions
include("adodb/toexport.inc.php");

// create an object instance
// configure library for a PostgreSQL connection
```

```
$db = NewADOConnection("postgres");

// open connection to database
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");

// execute query
$query = "SELECT title, id FROM library";
$result = $db->Execute($query) or die("Error in query: $query. " . $db-
>ErrorMsg());

// return a CSV string
echo rs2csv($result);

// close database connection
$db->Close();

?>
```

Podemos suprimir a primeira linha (nomes dos campos), adicionando um parâmetro para a chamada de rs2csv():

// return a CSV string

echo rs2csv($result, false);

Formatando com separador tab: Apenas troque a função (método) de rs2csv() para rs2tab().

Formatando a saída para tabela HTML: Para esta precisamos de outro importe, confira:

```
<html>
<head></head>
<body>
<?php
// uncomment this to see plaintext output in your browser
// header("Content-Type: text/plain");

// include the ADODB library
include("adodb/adodb.inc.php");

// include conversion functions
include("adodb/tohtml.inc.php");

// create an object instance
// configure it for a PostgreSQL connection
$db = NewADOConnection("postgres");

// open connection to database
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");

// execute query
$query = "SELECT titulo, id FROM biblioteca";
$result = $db->Execute($query) or die("Erro na consulta: $query. " . $db-
>ErrorMsg());

// return a table
```

```
echo rs2html($result);

// close database connection
$db->Close();

?>
</body>
</html>
```

Formatando Data de Saída

```
<?php
// Select a table, display the first two columns.
// If the second column is a date or timestamp, reformat the date to Brazilian
d/m/Y.

// uncomment this to see plaintext output in your browser
// header("Content-Type: text/plain");

// include the ADODB library
include("adodb/adodb.inc.php");

// create an object instance
// configure library for a PostgreSQL connection
$db = NewADOConnection("postgres");
// Antes alterar a tabela adicionando o campo data:
// \c dbbiblioteca
// ALTER TABLE biblioteca ADD COLUMN data date;

// open connection to database
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");

$recordSet = &$db->Execute('select id,data from biblioteca');
if (!$recordSet)
        print $db->ErrorMsg();
else
while (!$recordSet->EOF) {
        $fld = $recordSet->FetchField(1);
        $type = $recordSet->MetaType($fld->type);

        if ( $type == 'D' || $type == 'T')
                print $recordSet->fields[0].' '.
                        $recordSet->UserDate($recordSet-
>fields[1],'d/m/Y').'<BR>';
        else
                print $recordSet->fields[0].' '.$recordSet->fields[1].'<BR>';

        $recordSet->MoveNext();
}
$recordSet->Close(); # optional
$db->Close(); # optional

?>
```

Exporting in CSV or Tab-Delimited Format

```
<?php
// Exporting in CSV or Tab-Delimited Format
```

```
// uncomment this to see plaintext output in your browser
// header("Content-Type: text/plain");

include_once('adodb/toexport.inc.php');

// include the ADODB library
include("adodb/adodb.inc.php");

// create an object instance
// configure library for a PostgreSQL connection
$db = NewADOConnection("postgres");

// open connection to database
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");

$rs = $db->Execute('select titulo as "Título", autor as "Autor" from
biblioteca');

print "<pre>";
print rs2csv($rs); # return a string, CSV format

print '<hr>';

$rs->MoveFirst(); # note que alguns databases não suportam MoveFirst
print rs2tab($rs,false); # return a string, tab-delimited
# false == suppress field names in first line
print '<hr>';
$rs->MoveFirst();
rs2tabout($rs); # send to stdout directly (there is also an rs2csvout function)
print "

";
$rs->MoveFirst(); $fp = fopen($path, "w"); if ($fp) {
 rs2csvfile($rs, $fp); # write to file (there is also an rs2tabfile function)
 fclose($fp);

}
?>
```

INSERT com Arquivo SCV

```
<?php
// uncomment this to see plaintext output in your browser
// header("Content-Type: text/plain");

// include the ADODB library
include("adodb/adodb.inc.php");

// create an object instance
// configure library for a PostgreSQL connection
$db = NewADOConnection("postgres");

// open connection to database
```

```
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");

// prepare query
$query = $db->Prepare("INSERT INTO biblioteca (id, titulo, autor) VALUES (?, ?)
");

// read title-author list in from CSV file
$data = file("lista.txt");

// iterate through each line in file
foreach ($data as $l){
        // split on comma
        $arr = explode(",", $l);
        // insert values into prepared query
        $result = $db->Execute($query, array($arr[0], $arr[1])) or die("Error in
query: $query. " . $db->ErrorMsg());
}

// clean up
$db->Close;
echo "Arquivo CSV inserido com sucesso!";
?>
```

Arquivo lista.txt

```
Sandálias do Pescador, Morris West
Ana Karenina, Leon Tolstoi
Terras do Sem Fim, Jorge Amado
Helena, Machado de Assis
Paginação com muita Simplicidade
```

```
<?php
// Paginação com muita simplicidade

// uncomment this to see plaintext output in your browser
// header("Content-Type: text/plain");

// include the ADODB library
include("adodb/adodb.inc.php");

include_once('adodb/adodb-pager.inc.php');
session_start();


// create an object instance
// configure library for a PostgreSQL connection
$db = NewADOConnection("postgres");

// open connection to database
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");

$sql = "select * from biblioteca ";

$pager = new ADODB_Pager($db,$sql);
$pager->Render($rows_per_page=5);
```

```
?>
<pre>


<h2>Tratamento de Strings</h2>

<pre>
<?php
// Insert a row to the Orders table containing dates and strings that need to be
// quoted before they can be accepted by the database, eg: the single-quote in
the word John's.

// uncomment this to see plaintext output in your browser
// header("Content-Type: text/plain");

// include the ADODB library
include("adodb/adodb.inc.php");

// create an object instance
// configure library for a PostgreSQL connection
$db = NewADOConnection("postgres");

// open connection to database
$db->Connect("localhost", "postgres", "postabir", "dbbiblioteca") or die("Unable
to connect!");
$autor = $db->qstr("John's Old Shoppe"); // Tratamento de string com qstr antes
de inserir no banco

$sql = "insert into biblioteca (titulo,id,data,autor) ";
$sql .= "values ('Teste2',36,'2006-10-10',$autor)"; // Veja o original para
detalhes

if ($db->Execute($sql) === false) { // Só insere se não ocorrer erro
        print 'error inserting: '.$db->ErrorMsg().'<BR>';
}

echo "Inserido com sucesso!";
?>
```

Debugando

```
<?
include('adodb.inc.php');          # load code common to ADOdb
$conn = &ADONewConnection('access');        # create a connection
$conn->PConnect('northwind');   # connect to MS-Access, northwind dsn
$shipto = $conn->qstr("John's Old Shoppe");
$sql = "insert into orders (customerID,EmployeeID,OrderDate,ShipName) ";
$sql .= "values ('ANATR',2,".$conn->FormatDate(time()).",$shipto)";
$conn->debug = true;
if ($conn->Execute($sql) === false) print 'error inserting';
?>
```

Conectando-se a dois bancos de dados de SGBDs diferentes ao mesmo tempo

```
<?
include('adodb.inc.php');        # load code common to ADOdb
$conn1 = &ADONewConnection('mysql');  # create a mysql connection
$conn2 = &ADONewConnection('oracle');  # create a oracle connection

$conn1->PConnect($server, $userid, $password, $database);
$conn2->PConnect(false, $ora_userid, $ora_pwd, $oraname);

$conn1->Execute('insert ...');
$conn2->Execute('update ...');
?>
```

Recordset Filters

Sometimes we want to pre-process all rows in a recordset before we use it. For example, we want to ucwords all text in recordset.

```
include_once('adodb/rsfilter.inc.php');
include_once('adodb/adodb.inc.php');
// ucwords() every element in the recordset
function do_ucwords(&$arr,$rs)
{
        foreach($arr as $k => $v) {
                $arr[$k] = ucwords($v);
        }
}

$db = NewADOConnection('mysql');
$db->PConnect('server','user','pwd','db');

$rs = $db->Execute('select ... from table');
$rs = RSFilter($rs,'do_ucwords');
```

Executing the SQL

```
$result = $db->Execute("SELECT * FROM employees");
if ($result === false) die("failed");
```

Inserts and Updates

Let's say you want to insert the following data into a database.

```
ID = 3
TheDate=mktime(0,0,0,8,31,2001) /* 31st August 2001 */
Note= sugar why don't we call it off

$sql = "INSERT INTO table (id, thedate,note) values ("
   . $ID . ','
   . $db->DBDate($TheDate) .','
   . $db->qstr($Note).")";
$db->Execute($sql);
```

MetaTypes

You can find out more information about each of the fields (I use the words fields and columns interchangebly) you are selecting by calling the recordset method FetchField($fieldoffset). This will return an object with 3 properties: name, type and max_length.

For example:

$recordset = $conn->Execute("select adate from table");

$f0 = $recordset->FetchField(0);

Then $f0->name will hold 'adata', $f0->type will be set to 'date'. If the max_length is unknown, it will be set to -1.

One problem with handling different databases is that each database often calls the same type by a different name. For example a timestamp type is called datetime in one database and time in another. So ADODB has a special MetaType($type, $max_length) function that standardises the types to the following:

```
C: character and varchar types
X: text or long character (eg. more than 255 bytes wide).
B: blob or binary image
D: date
T: timestamp
L: logical (boolean)
I: integer
N: numeric (float, double, money)
```

In the above date example,

```
$recordset = $conn->Execute("select adate from table");
$f0 = $recordset->FetchField(0);
$type = $recordset->MetaType($f0->type, $f0->max_length);
print $type; /* should print 'D' */
```

Apresentando a adobDB

Bom artigo do Silas Alves Júnior

http://www.revistaphp.com.br/print.php?id=49

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Abstra%C3%A7%C3%B5es_de_Bancos_de_Dados/AdoDB"
Page categories: PHP

Aplicativos em PHP/Abstrações de Bancos de Dados/PEAR/DB

Conteúdo

- 1 Abstração PEAR/DB
- 2 PEAR/DB Exemplos Testados
- 2.1 Conexão
- 2.2 Query
- 2.3 fetchInto
- 2.4 fetchRow
- 2.5 fetchInfo
- 2.6 fetchInto
- 2.7 Número de Registros
- 2.8 Número de Campos
- 2.9 afectedRows
- 2.10 TableInfo
- 2.11 Prepare e Execute
- 2.12 Prepare e Execute (arrays)
- 2.13 Passando arrays para Execute
- 2.14 Usando executeMultiple ao invés de execute
- 2.15 AutoPrepare
- 2.16 Outro
- 2.17 AutoExecute
- 2.18 Outro
- 2.19 Tornando ON todas as opções de Portabilidade enquanto conecta
- 2.20 Desconectar
- 2.21 GetAll
- 2.22 GetAssoc - ret array
- 2.23 getCol - dados de coluna
- 2.24 Retorna primeira linha da primaira coluna
- 2.25 Outro
- 2.26 Retorna Primeiro Registro
- 2.27 limit
- 2.28 Free
- 2.29 DbError
- 2.30 Referências

Abstração PEAR/DB

Este pacote da PEAR foi substituído pelo MDB mas continua sendo mantido em termos de bugs e segurança.

É uma abstração de bancos de dados com boas funções que tanto economizam o trabalho de desenvolvimento quanto facilitam a migração entre os diversos SGBDs suportados: fbsql, ibase, informix, msql, mssql, mysql, mysqli, oci8, odbc, pgsql, sqlite and sybase.

PEAR/DB Exemplos Testados

A PEAR/DB é uma abstração de Bancos de Dados para uso com o PHP. Abstração é uma técnica que simplifica algo complexo.

Obs.: Nem todos os SGBDs suportam todos os métodos e propriedades da PEAR/DB.

Conexão

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
// Forma genérica: $db =&
DB::connect('tiposgbd://$usuario:$senha@$host:$porta/$banco');

require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}
```

Query

```
// Proceed with getting some data...
$res =& $db->query('SELECT * FROM clientes');
```

fetchInto

```
// Get each row of data on each iteration until
// there are no more rows
//while ($res->fetchInto($row)) {
    // Assuming DB's default fetchmode is DB_FETCHMODE_ORDERED
//    echo $row[0] . "<br>";
//}
```

fetchRow

```
// Or, you could have done the same thing using fetchRow()
 while ($row =& $res->fetchRow()) {
     // Assuming DB's default fetchmode is DB_FETCHMODE_ORDERED
     echo $row[0] . "<br>";
```

```
 }
?>
```

fetchInfo

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}
// Once you have a valid DB object named $db...
$res =& $db->query('SELECT * FROM clientes');

while ($res->fetchInto($row, DB_FETCHMODE_ASSOC)) {
    echo $row['codigo'] . "\n";
}
?>
```

fetchInto

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$db->setFetchMode(DB_FETCHMODE_ASSOC);

$res =& $db->query('SELECT * FROM clientes');

while ($res->fetchInto($row)) {
    echo $row['nome'] . "\n";
}
?>
```

Número de Registros

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}
```

```
// Once you have a valid DB object named $db...
$res =& $db->query('SELECT * FROM clientes');
echo $res->numRows();

$res->free();
?>
```

Número de Campos

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$res =& $db->query('SELECT * FROM clientes');
echo $res->numCols();

$res->free();
?>
```

afectedRows

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// remember that this statement won't return a result object. Para: INSERT,
UPDATE or DELETE
$db->query('DELETE * FROM clientes');
echo 'I have deleted ' . $db->affectedRows() . ' clients';

?>
```

TableInfo

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
```

```
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$res =& $db->query('SELECT * FROM clientes');
print_r($db->tableInfo($res));

// That usage works for DB 1.6.0 or later.
// Below is the syntax for earlier versions:
print_r($res->tableInfo());

$res->free();
?>
```

Prepare e Execute

<?php // Create a valid DB object named $db // at the beginning of your program... require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes'); if (PEAR::isError($db)) {

```
   die($db->getMessage());
```

}

// Once you have a valid DB object named $db... // Inserir dois registros apenas com o campo código $sth = $db->prepare('INSERT INTO clientes (codigo) VALUES (?)'); $db->execute($sth, 15000); $db->execute($sth, 18000);

//$res->free(); ?>

Prepare e Execute (arrays)

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...(Adiciona um registro - 15001,
com 3 campos)
$sth = $db->prepare('INSERT INTO clientes VALUES (?, ?, ?)');

$data = array(15001, 'quinze', 'amail15');
$db->execute($sth, $data);

//$res->free();
?>
```

Passando arrays para Execute

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$alldata = array(array(15002, 'quinze2', 'enemail'),
                 array(15003, 'quinze3', 'toemail'),
                 array(15004, 'quinze4', 'treemail'),
                 array(15005, 'quinze5', 'fireemail'));
$sth = $db->prepare('INSERT INTO clientes VALUES (?, ?, ?)');

foreach ($alldata as $row) {
    $db->execute($sth, $row);
}
?>
```

Usando executeMultiple ao invés de execute

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$alldata = array(array(16001, 'one', 'en'),
                 array(16002, 'two', 'to'),
                 array(16003, 'three', 'tre'),
                 array(16004, 'four', 'fire'));
$sth = $db->prepare('INSERT INTO clientes VALUES (?, ?, ?)');
$db->executeMultiple($sth, $alldata);

?>
```

AutoPrepare

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
```

```
    die($db->getMessage());
}

// With autoPrepare(), you don't have to write your insert or update queries.
For example:
// Once you have a valid DB object named $db...
$table_name   = 'clientes';
$table_fields = array('codigo', 'nome', 'email');

$sth = $db->autoPrepare($table_name, $table_fields, DB_AUTOQUERY_INSERT);

$table_values = array(16006, 'Fabien', 'France@');

$res =& $db->execute($sth, $table_values);

if (PEAR::isError($sth)) {
    die($sth->getMessage());
}
?>
```

Outro

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$table_name   = 'clientes';
$table_fields = array('nome', 'email');
$table_values = array('Roberto', 'Brasil@');

$sth = $db->autoPrepare($table_name, $table_fields, DB_AUTOQUERY_UPDATE, 'codigo
= 5600');

if (PEAR::isError($sth)) {
    die($sth->getMessage());
}

$res =& $db->execute($sth, $table_values);

if (PEAR::isError($res)) {
    die($res->getMessage());
}

?>
```

AutoExecute

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$table_name = 'clientes';

$fields_values = array(
    'codigo'      => 16007,
    'nome'    => 'Fabien',
    'email' => 'France@'
);

$res = $db->autoExecute($table_name, $fields_values, DB_AUTOQUERY_INSERT);

if (PEAR::isError($res)) {
    die($res->getMessage());
}

?>
```

Outro

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$db =& DB::connect('pgsql://postgres:senha@localhost:5432/testes');
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$table_name = 'clientes';

$fields_values = array(
    'nome'    => 'Fabien2',
    'email' => 'France2'
);

$res = $db->autoExecute($table_name, $fields_values, DB_AUTOQUERY_UPDATE,
'codigo = 1234');

if (PEAR::isError($res)) {
    die($res->getMessage());
}

?>
```

Tornando ON todas as opções de Portabilidade enquanto conecta

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

?>
```

Desconectar

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

$db->disconnect();
?>
```

GetAll

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
```

```
    die($db->getMessage());
}

// Retorna um array de dados
$db->setFetchMode(DB_FETCHMODE_ASSOC);
$data =& $db->getAll('SELECT codigo, nome, email FROM clientes');

if (PEAR::isError($data)) {
    die($data->getMessage());
}

print_r($data);

?>
```

GetAssoc - ret array

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Retorna um array de dados
// Once you have a valid DB object named $db...
$data =& $db->getAssoc('SELECT codigo, nome FROM clientes');

if (PEAR::isError($data)) {
    die($data->getMessage());
}

print_r($data);

?>
```

getCol - dados de coluna

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
```

```
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Retorna dados de uma coluna
$data =& $db->getCol('SELECT codigo, nome FROM clientes');

if (PEAR::isError($data)) {
    die($data->getMessage());
}

print_r($data);
?>
```

Retorna primeira linha da primaira coluna

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Retorna primeira linha da primaira coluna
$data =& $db->getOne('SELECT nome FROM clientes');

if (PEAR::isError($data)) {
    die($data->getMessage());
}

echo "$data\n";?>
```

Outro

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
```

```
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

$data =& $db->getOne('SELECT nome FROM clientes WHERE codigo IN (?, ?)',
array(3, 7));

if (PEAR::isError($data)) {
    die($data->getMessage());
}

echo "$data\n";

?>
```

Retorna Primeiro Registro

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Once you have a valid DB object named $db...
$data =& $db->getRow('SELECT codigo, nome FROM clientes',
        array(), DB_FETCHMODE_ORDERED);

if (PEAR::isError($data)) {
    die($data->getMessage());
}

print_r($data);

?>
```

limit

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// Limit de, conte
$res =& $db->limitQuery('SELECT * FROM clientes', 1100, 30);

echo $res->numRows();

if (PEAR::isError($res)) {
    die($res->getMessage());
}

?>
```

Free

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

$res =& $db->query('SELECT nome, email FROM clientes');
while ($row =& $res->fetchRow()) {
    echo $row[1] . ', ' . $row[2] . "\n";
}
$res->free();

?>
```

DbError

```
<?php
// Create a valid DB object named $db
// at the beginning of your program...
require_once 'DB.php';

$dsn='pgsql://postgres:senha@localhost:5432/testes';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    /*
     * This is not what you would really want to do in
     * your program.  It merely demonstrates what kinds
     * of data you can get back from error objects.
     */
    echo 'Messagem Padrão: ' . $db->getMessage() . "\n";
    echo 'Código Padrão: ' . $db->getCode() . "\n";
    echo 'DBMS/User Message: ' . $db->getUserInfo() . "\n";
    echo 'DBMS/Debug Message: ' . $db->getDebugInfo() . "\n";
    exit;
}

?>
```

Referências

http://pear.php.net

http://vulcanonet.com/soft/?pack=pear_tut

http://www.phpbuilder.com/columns/allan20010115.php3?print_mode=1

Estes exemplos foram testados com:

Apache 2.2

PHP 5.1 (ambos no Xampp 1.5.1)

PostgreSQL 8.1.2

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Abstra%C3%A7%C3%B5es_de_Bancos_de_Dados/PEAR/DB"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Formulários

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Excluir Registros através de CheckBox

Temos a seguinte situação: ao listar registros de uma tabela queremos selecionar alguns deles através de um checkbox e excluir os selecionados.

```
<?php
/*
Banco - excluir_varios
Tabela
CREATE TABLE `produtos` (
  `id` int(11) NOT NULL,
  `produto` char(45) default NULL,
  `categoria` char(45) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
*/

$conexao = mysql_connect('localhost','root','');
mysql_select_db('excluir_varios',$conexao);
$consultar = "SELECT * FROM produtos ORDER BY id";
$resultado = mysql_query($consultar, $conexao);

if(mysql_num_rows($resultado) != 0){
        echo "<form name='frmExcluir' method='post'
action='excluir_varios.php'>";
        echo "<table
border=1><tr><th> </th><th>Produto:</th><th>Categoria:</th></tr>";

        while($linha = mysql_fetch_row($resultado)){
                echo "<td><input type='checkbox' name='id[]'
value='$linha[0]'></td>
                        <td>$linha[1]</td>
                        <td>$linha[2]</td></tr>";
        }
        echo "<tr><td colspan='3'><input type='submit' name='excluir'
value='Excluir!'></td></tr>";
        echo "</table></form>";
}else{
        echo "Nenhum registro foi encontrado!";
}
if(isset($_POST['id'])){
        $opcoes = $_POST['id'];
        $opcoes_text = implode(", ", $opcoes);
```

```
        $strexcluir = "DELETE FROM produtos WHERE id in (" . $opcoes_text . ")";
        mysql_query($strexcluir, $conexao) or die("Ocorreu algum erro");
}else{
        echo "É necessário escolher quem será excluído<br>";
        echo "<a href='javascript: history.back();'>Voltar</a>";
}
?>
```

Manipulando dados de formulários com PHP do Alfred Reinold Baudisch

Parte 1 - http://www.linhadecodigo.com.br/artigos.asp?id_ac=220

Parte 2 - http://www.linhadecodigo.com.br/artigos.asp?id_ac=244

Formulários acessíveis à prova de spam

http://www.linhadecodigo.com.br/artigos_impressao.asp?id_ac=1284

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Formul%C3%A1rios"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/URLs

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

11.2 - Trabalhando com URL em PHP

Passando Parâmetros pela URL

Primeiro (âncora)

<a href="arquivo.php?parametro1=valor1&parametro2=valor2&parametro3=valor3">Link</a>

```
arquivo.php
$par1=$_GET['parametro1'];
$par2=$_GET['parametro2'];
$par3=$_GET['parametro3'];
```

Segundo (action de form)

```
<form name=frm method=post
action="arquivo2.php?parametro1=valor1&parametro2=valor2">
...
```

arquivo2.php

```
$par1=$_POST['parametro1'];
$par2=$_POST['parametro1'];
```

Terceiro (URL)

http://localhost/teste.php?parametro1=valor1

teste.php

$par1=$_GET['parametro1'];

Quarto (location no javascript)

```
<?php
// Já vindo de outro script, chamado via POST
$a = $_POST['a'];

?>
<script>
if(confirm("Confirma?")){
        location="vai.php?a='<?=$a?>' ";
```

```
}else{
        location='volta.php';
}
</script>
```

Reconstruct URL string in PHP

```
 // find out the domain:
 $domain = $_SERVER['HTTP_HOST'];
 // find out the path to the current file:
 $path = $_SERVER['SCRIPT_NAME'];
 // find out the QueryString:
 $queryString = $_SERVER['QUERY_STRING'];
 // put it all together:
 $url = "http://" . $domain . $path . "?" . $queryString;
 echo "The current URL is: " . $url . "
";

 // An alternative way is to use REQUEST_URI instead of both
 // SCRIPT_NAME and QUERY_STRING, if you don't need them seperate:
 $url2 = "http://" . $domain . $_SERVER['REQUEST_URI'];
 echo "The alternative way: " . $url2;
```

Do site - http://snippets.dzone.com/posts/show/4054

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/URLs"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Funções

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Funções são blocos de código, conjuntos de instruções, que ficam quietos, sem vida, até que sejam chamadas em algum lugar do código. Funções reduzem o trabalho de digitação, o tamanho dos scripts em geral, os erros, reaproveitando código, facilitando e organizando o trabalho de desenvolvimento.

Conteúdo

- 1 Exemplos de funções definidas pelo usuário
- 2 Variáveis globais
- 3 Variável externa acessível na função
- 4 Acessando variáveis Externas
  - 4.1 Variável externa inacessível
- 5 Variáveis estáticas
- 6 Recursiva
- 7 Declarando variáveis static
- 8 Retornar mais de um valor de uma função
- 9 Passando Argumentos através de Funções
- 10 Por Valor
- 11 Por Referência
- 12 Otimização do tempo de execução
- 13 Função com parâmetro default
- 14 Trabalhando com Funções OB

Exemplos de funções definidas pelo usuário

```
function quadrado($numero) {
        print $numero*$numero;
}
```

Executando:

quadrado(6); // Saída: 36

Variáveis globais

```
$var1 = 5;
function testeGlobal1(){
        $var1 = 1;
        print "<br>Valor de \$var1: $var1";
}
echo testeGlobal1();
```

Variável externa acessível na função

```
$var2 = 10;
function testeGlobal2(){
        global $var2;
        print "<br>Valor de \$var2 $var2";
}
echo testeGlobal2();

$var5 = 15;
function testeGlobal5(){
        $var5 = 5;
    print "<br><br>A variável global vale $GLOBALS[var5], ";
    print "Já a variável local vale $var5<br><br>";
}

testeGlobal5();

function cliente($codigo, $idade = 18){
        print "Código = $codigo, Idade = $idade";
}

cliente(1);  //Exibirá: Código = 1, Idade = 18

function cubo($num){
        return ($num*$num*$num);
}

$var1 = 2 * cubo(5);echo "<br>".$var1;
```

Acessando variáveis Externas

Normalmente de dentro de uma função não temos acesso às variáveis que estão fora dela. Mas veja como contornar isso:

Variável externa inacessível

```
$var1 = 5;
function testeGlobal1(){
        $var1 = 1;
        print ”Valor de \$var1: $var1”;
}
echo testeGlobal1();
Variável externa acessível na função

$var2 = 10;
function testeGlobal2(){
        global $var2;
        print ”Valor de \$var2 $var2”;
}
echo testeGlobal2();
```

Outra alternativa é usar o array $GLOBALS[], que contém todas as variáveis globais. Veja um exemplo:

```
$var5 = 15;
function testeGlobal5(){
        $var5 = 5;
        print "<br><br>A variável global vale $GLOBALS[var5], ";
        print "Já a variável local vale $var5<br><br>";
}

testeGlobal5();
```

Variáveis estáticas

```
function contador(){
        static $x = 0;
        return $x++;
}
echo "<br>";
echo contador();echo contador();echo contador();
//A saída será: 012

function contador2(){
        $x = 0;
        return $x++;
}
echo "<br>";
echo contador2();echo contador2();echo contador2();
//A saída será: 000.

  function staticfunction() {
   static $count = 0;
   $count++;
        if ($count==1){
```

```
       echo "A Função foi executada $count vez<br>";
        }else{
           echo "A Função foi executada $count vezes<br>";
        }
  }

  for($i = 0; $i < 5; $i++) {
   staticfunction();
  }

function Teste1()
{
    static $a = 0;
    echo $a;
    $a++;
}

for($x=0;$x<=10;$x++){
        echo Teste1()." ";
}
```

Recursiva

```
function Teste()
{
    static $count = 0;

    $count++;
    echo $count." ";
    if ($count < 10) {
        Teste ();
    }
    $count--;
}

Teste();
```

Declarando variáveis static

```
function foo(){
    static $int = 0;          // correro
   // static $int = 1+2;        // errado (é uma expressão)
   // static $int = sqrt(121);  // errado  (é uma expressão também)

    $int++;
    echo $int;
}


function aumentoSalario($sal, $perc=5){
        $salario = $sal * $perc/100;
        echo $salario;
}
echo "<br>Aumento: " . aumentoSalario(1956);
```

```
function redirecionar($url){
        header("Location: $url");
}
echo "<br>";
redirecionar("http://ribafs.phpnet.us/");
echo "<br>";
```

Retornar mais de um valor de uma função

usa-se arrays e list()

array() retorna e list() exibe

//Exemplo:

```
function recebe_hoje(){
        $data_abreviada=date("d/m/Y");
        $data_extensa=date("l, d F \d\e Y");

        return array($data_abreviada, $data_extensa);
}

list($data_abreviada, $data_extensa)=recebe_hoje();
        print $data_extensa;
echo "<br>";
        print $data_abreviada;


<h2>Declaração dinâmica de função</h2>

<pre>
if ($f == 1){
        function f1(){
                echo "funcao1";
        }
}else{
        function f2(){
                echo "funcao2";
        }
}

<h2>Retornando o número de argumentos de uma função</h2>

<pre>
function ret_args_funcao() {
    $numargs = func_num_args();
    echo "Número de argumentos: $numargs<br>\n";
    if ($numargs >= 2) {
    echo "Segundo argumento vale : " . func_get_arg (1) . "<br>\n";
    }
    $arg_list = func_get_args();
    for ($i = 0; $i < $numargs; $i++) {
    echo "Argumento $i vale: " . $arg_list[$i] . "<br>\n";
    }
}

ret_args_funcao (1, 2, 3);
```

Passando Argumentos através de Funções

O default é 'por valor', que passa uma cópia do valor da variável.

Também podemos passar 'por referência', onde se passa o endereço da própria variável. Quando atribuímos uma variável a outra passando como referência, não fazemos cópia alguma, mas sim passamos o endereço da variável original, portanto qualquer alteração nesta referência refletirá na variável original.

ByRef é mais eficiente em termos de memória ao lidar com variáveis e arrays grandes e também permite alterar o valor da variável original, o que não acontece com o ByVal, mas a vantagem de desempnho somente é percebida em grandes arrays em grandes loops.

Para maior segurança setar:

allow_call_time_pass_reference no php.ini

Impede a passagem de valores por referência nas chamadas, mas permite somente na definição das funções.

$var1 = & $var2;

Agora ambas apontam para o mesmo endereço e valor.

Reter valor de variáveis entre chamadas (static)

(Guarda o valor da última chamada até o final da execução do script, tantas vezes quantas a função for chamada).

Exemplo:

```
$valor = 4;
$ref = &$valor;

$ref = 3;

$valor = 4;
$ref = &$valor; // Aqui tornamos ambas as variáveis com o mesmo endereço
                // O que alterarmos em $ref alteramos em $valor

$ref = 3;       // Com isso também alteramos $valor para 3, veja abaixo.
echo $valor . "<br>";

$valor=0;          // Com isso também alteramos $ref para 0, veja abaixo.

echo $ref;
```

Por Valor

```
function val_subtracao($num1, $num2){
        if($num1 < $num2){
                die("Números negativos");
        }else{
                $return_result=0;
                while($num1 > $num2){
```

```
                        $num1 = $num1 - 1;
                        $return_result = $return_result + 1;
                }
        }
        return($return_result);
}

$primeiro_op=493;
$segundo_op=355;
$resultado1 = val_subtracao($primeiro_op, $segundo_op);
print ("Resultado1 é $resultado1<br>");
$resultado2 = val_subtracao($primeiro_op, $segundo_op);
print("Resultado2 é $resultado2<br>");
```

Por Referência

```
function subtracao_ref(&$num1, &$num2){
        if($num1 < $num2){
                die("Números negativos");
        }else{
                $return_result=0;
                while($num1 > $num2){
                        $num1 = $num1 - 1;
                        $return_result = $return_result + 1;
                }
        }
        return($return_result);
}

$primeiro_op=493;
$segundo_op=355;
$resultado1 = subtracao_ref($primeiro_op, $segundo_op);
print ("<br><br>Resultado1 é $resultado1<br>");
$resultado2 = subtracao_ref($primeiro_op, $segundo_op);
print("Resultado2 é $resultado2<br>");

echo "Agora, se se nós executarmos exatamente a mesma chamada da subtração como
fizemos a primeira vez, receberemos a saída:
resultado1 é 138 e resultado2 é 0";


/*
Sugestão de chamada de função:

if (nome_funcao($argumento){
        ....
        ....
}else{
        ....
}

*/
?>

<?php
// Retorna o tipo e o valor de variável
function ss_as_string (&$thing, $column = 0) {
```

```
    if (is_object($thing)) {
        return ss_object_as_string($thing, $column);
    }
    elseif (is_array($thing)) {
        return ss_array_as_string($thing, $column);
    }
    elseif (is_double($thing)) {
        return "Double(".$thing.")";
    }
    elseif (is_long($thing)) {
        return "Long(".$thing.")";
    }
    elseif (is_string($thing)) {
        return "String(".$thing.")";
    }
    else {
        return "Unknown(".$thing.")";
    }
}

// Retorna o tipo e o valor de array
function ss_array_as_string (&$array, $column = 0) {
    $str = "Array(<BR>\n";
    while(list($var, $val) = each($array)){
        for ($i = 0; $i < $column+1; $i++){
            $str .= "    ";
        }
        $str .= $var.' ==> ';
        $str .= ss_as_string($val, $column+1)."<BR>\n";
    }
    for ($i = 0; $i < $column; $i++){
        $str .= "    ";
    }
    return $str.')';
}

// Retorna o tipo e o valor de objeto
function ss_object_as_string (&$object, $column = 0) {
    if (empty($object->classname)) {
        return "$object";
    }
    else {
        $str = $object->classname."(<BR>\n";
        while (list(,$var) = each($object->persistent_slots)) {
            for ($i = 0; $i < $column; $i++){
                $str .= "    ";
            }
            global $$var;
            $str .= $var.' ==> ';
            $str .= ss_as_string($$var, column+1)."<BR>\n";
        }
        for ($i = 0; $i < $column; $i++){
            $str .= "    ";
        }
        return $str.')';
    }
}

$var="Riba";
```

```
echo ss_as_string($var);
//echo ss_as_string($GLOBALS);
```

Otimização do tempo de execução

```
function ss_timing_start ($name = 'default') {
    global $ss_timing_start_times;
    $ss_timing_start_times[$name] = explode(' ', microtime());
}

function ss_timing_stop ($name = 'default') {
    global $ss_timing_stop_times;
    $ss_timing_stop_times[$name] = explode(' ', microtime());
}

function ss_timing_current ($name = 'default') {
    global $ss_timing_start_times, $ss_timing_stop_times;
    if (!isset($ss_timing_start_times[$name])) {
        return 0;
    }
    if (!isset($ss_timing_stop_times[$name])) {
        $stop_time = explode(' ', microtime());
    }
    else {
        $stop_time = $ss_timing_stop_times[$name];
    }
    // do the big numbers first so the small ones aren't lost
    $current = $stop_time[1] - $ss_timing_start_times[$name][1];
    $current += $stop_time[0] - $ss_timing_start_times[$name][0];
    return $current;
}

?>
```

Função com parâmetro default

Parâmetro default é aquele cujo valor já faz parte da função, sendo opcional.

```
function cliente($codigo, $idade = 18){
        print "Código = $codigo, Idade = $idade";
}

cliente(1);  //Exibirá: Código = 1, Idade = 18
```

Trabalhando com Funções OB

- Artigo do Adriano Oliveira Gonçalves na Revista do PHP

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Fun%C3%A7%C3%B5es"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Constantes mágicas e super globais

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 Variáveis do servidor
- 2 Exemplos
- 3 Constantes Mágicas
- 4 Tutorial Sobre as Super Globais

Variáveis do servidor

$_SERVER

Este é um array (vetor) 'superglobal', ou automaticamente global. Isto significa que ele é disponível em todos os escopos (níveis) de um script. Você não precisa fazer um: ... global $_SERVER; ... para poder acessá-lo dentro de funções ou métodos, como era necessário com $HTTP_SERVER_VARS. O array superglobal $_SERVER existe em qualquer sessão PHP e já contém um conjunto de chaves (índices) pré definidos e valorados. Os índices mais importantes são:

'REQUEST_URI'

O URI fornecido para acessar a página atual, por exemplo, '/index.html'.

'SCRIPT_NAME'

Contém o caminho completo do script atual. Útil para páginas que precisam apontar para elas mesmas (dinamicamente). A constante __FILE__ contém o caminho completo e nome do arquivo (mesmo incluído) atual.

'PHP_SELF'

O nome do arquivo do script atualmente em uso, relativo ao document root. Por exemplo, $_SERVER['PHP_SELF'] em um script com o endereço http://example.com/test.php/foo.bar pode ser /test.php/foo.bar. A constante __FILE__ contém o caminho completo e nome do arquivo (mesmo incluído) atual.

Se estiver rodando o PHP em linha de comando, esta variável não está disponível.

'SERVER_NAME'

O nome host do servidor onde o script atual é executado. Se o script está rodando em um host virtual, este será o valor definido para aquele host virtual.

'REQUEST_METHOD'

Contém o método de request utilizando para acessar a página. Geralmente 'GET', 'HEAD', 'POST' ou 'PUT'.

'QUERY_STRING'

A query string (string de solicitação), se houver, pela qual a página foi acessada.

'DOCUMENT_ROOT'

O diretório raiz sob onde o script atual é executado, como definido no arquivos de configuração do servidor.

'SCRIPT_FILENAME'

O caminho absoluto o script atualmente em execução.

Nota: Se o script for executado pela CLI com um caminho relativo, como file.php ou ../file.php, $_SERVER['SCRIPT_FILENAME'] irá conter o caminho relativo especificado pelo usuário.

Exemplos

```
$current_script = dirname($_SERVER['SCRIPT_NAME']);

$current_path  = dirname($_SERVER['SCRIPT_FILENAME']);

$request_uri = $_SERVER['REQUEST_URI'];

// Pick the predefined variable that works on your server
     return $_ENV['SCRIPT_URL'];

$_SERVER['QUERY_STRING'])

      $sPathPS = $_SERVER[PHP_SELF];
       $sPathFS = __FILE__;

echo 'http';
if($_SERVER['HTTPS']=='on'){echo 's';}
echo '://'.$_SERVER['SERVER_PORT'].$_SERVER['SCRIPT_NAME'];
if($_SERVER['QUERY_STRING']>' '){echo '?'.$_SERVER['QUERY_STRING'];}
```

Constantes Mágicas

```
__LINE__          A linha atual do script.
__FILE__         O caminho completo e nome do arquivo. Se utilizado dentro de um
include, o nome do arquivo incluído será retornado.
__FUNCTION__     O nome da função (Acrescentado no PHP 4.3.0). A partir do PHP 5
esta constante retorna o nome da função como ela foi
   declarada (sensível a maiúsculas e minúsculas). No PHP 4 sempre retorna o
nome em minúsculas.
__CLASS__        O nome da classe (Acrescentado no PHP 4.3.0). A partir do PHP 5
esta constante retorna o nome da função como ela foi
   declarada (sensível a maiúsculas e minúsculas). No PHP 4 sempre retorna o
nome em minúsculas.
__METHOD__       O nome do método de classe. (Acrescentado no PHP 5.0.0). O nome
do método é retornado como foi declarado (sensível a
```

```
   maiúsculas e minúsculas).

if (realpath(__FILE__) == realpath($_SERVER['SCRIPT_FILENAME'])) {
  exit;
}
```

Tutorial Sobre as Super Globais

Do Pedro Henrique da Silva Gregório no Vivaolinux -

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Constantes_m%C3%A1gicas_e_super_globais"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Session

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

11.5 - Trabalhando com Session em PHP

Crie um pequeno aplicativo com os seguintes arquivos. Copie os arquivos para um diretório no seu servidor web e chame o index pelo browser:

index.php

```
<?
session_start();
?>
<center><h1>Trabalhando com SESSION em PHP</h1>
Podemos preservar valores de variáveis enquanto durar uma sessão do browser
através do uso de
SESSION.<br>
Para isso devemos startar a sessão em cada página que desejamos usar esta
variável com<br>
<br>
sesssion_start();<br><br>
Lembrando que esta função deve vir antes de qualquer comando que mande algo para
a tela, caso
<br>
o session esteja configurado para usar cookie.<br>
Na primeira página deve ter um formulário com algum campo que devemos usar no
session.<br>
Experimente gravar a URL de uma das páginas internas e acessar diretamente
(http://localhost/session3)
<br>Primeiro feche todas as seções do browser e depois abra o browser com essa
URL.<br>
<br>
Veja que SESSION é muito bom para preservar o valor de variáveis entre páginas
de um site numa seção.
<br>
Portanto seu uso é muito útil quando pretendemos autenticas os visitantes de
todas as as páginas
de um site.<BR>

Como também para outros usos em que pretendemos reaproveitar o valor de
variáveis
(algo como global).<BR>
Acompanhe este exemplo para ver detalhes.<br><br><br>

<form method=post action=session2.php>
        Login<input type=text size=8 name=login><br>
```

```
        <input type=submit value=Enviar>
</form></center>
```

destruir.php

```
<?
session_start();
if (isset($_SESSION['login'])){
  session_unset(); // Eliminar todas as variáveis da sessão
  session_destroy(); // Destruir a sessão
  echo "Entre. Sessão Destruida. <a href=session3.php>Session3</a><br>";
} else {
  echo "Acesso não autenticado!";
}
?>
```

session2.php

```
<?
session_start();
$_SESSION['login']=$_POST['login'];

if (isset($_SESSION['login'])){
        echo "Entre. Session2. <a href=session3.php>Session3</a>";
} else {
        echo "Acesso não autenticado!";
}
?>
```

session3.php

```
<?
session_start();
if (isset($_SESSION['login'])){
  echo "Entre. Session3. <a href=session4.php>Session4</a><br>";
  echo "<a href=destruir.php>Drestruir Sessão</a>";
} else {
  echo "Acesso não autenticado!";
}
?>
```

session4.php

```
<?
session_start();
if (isset($_SESSION['login'])){
        echo "Entre. Session4. <a href=session5.php>Session5</a>";
} else {
        echo "Acesso não autenticado!";
}
?>
```

session5.php

```
<?
session_start();
if (isset($_SESSION['login'])){
        echo "Entre. Session5. <a href=index.php>Index</a><br><br>";
        echo "Informações: <br>ID da Sessão: <b>" . session_id() .
        "</b><br>Variável mantida pela SuperGlobal \$_SESSION: <b>" .
$_SESSION['login'];
} else {
        echo "Acesso não autenticado!";
}
?>
```

Como as variaveis ficam fixas em todas as páginas de um site.Um exemplo: Quando você faz login em um site ao qual tem um cadastro.E seu nome de login fica lá em cima em todas as páginas referente a sua conta.Essas variaveis que "vivem" em todo o site são chamadas "variaveis de sessão($_SESSION). Para cria-las é muito simples:

primeiro devemos começar sempre antes de qualquer tag HTML ou escapamento do comando echo("");

ficaria assim um exemplo:

```
<?php

     session_start(); // sempre chamamos a sessão dessa forma

     $login="cloud"; // passamos um valor qualquer a variavel $login "ainda não
é a sessao"

     $_SESSION['login']=$login;
?>
```

agora sim,criamos uma variavel session que tem o nome login(....['login']) e ela possue agora um valor que é da variavel $login(essa variavel $login não precissa necessariamente ter um valor fixo,ela pode vir de um form entenderam?

quando quisermos mostrar ou usar o valor da variavel $login em outra pagina,ja podemos pois agora ela esta dentro de uma session um exemplo de como chama-la em outra pagina(arquivo)

```
<?php

     session_start(); // sempre que usarmos as sessions devemos chamar esse
codigo sempre no inicio do script

if(isset($_SESSION['login'])){// verifica se existe a varavel session

   $login=$_SESSION['login']; // passa o valor da variavel session para outra
variavel so que uma variavel dentro do mesmo arquivo

   echo($login);// cloud

}else{

  echo("vc nao passou pelo arquivo anterior");
```

}

Capítulo sobre Session na Documentação Oficial do PHP

http://www.php.net/manual/pt_BR/ref.session.php

Artigos Sobre Session em PHP

http://www.revistaphp.com.br/print.php?id=79

http://www.scriptbrasil.com.br/forum/lofiversion/index.php/t79221.html

http://www.htmlstaff.org/ver.php?id=5353

http://maozinhadaweb.blogspot.com/2007/05/tutorial-de-php-parte-3-sesses-e.html

http://www.devshed.com/c/a/PHP/Sessions-and-Cookies/

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Session"
Page categories: PHP

Trabalhando em PHP com/Arquivos

Conteúdo

- 1 Abrir arquivo
- 2 Gravar em Arquivo
- 3 Ler Arquivo
- 4 fgets()
- 5 fgetc()
- 6 Ler Todo um Arquivo
- 7 file()
- 8 Ler Arquivo Via URL
- 9 feof(handle)
- 10 Contando o Número de Linhas de um arquivo
- 11 Contar palavras de um arquivo mostrando duplicadas
- 12 Ler de forma inversa um arquivo, linha a linha
- 13 Ler aleatoriamente linha de arquivo
- 14 Ler linha específica de arquivo
- 15 Operações com Diretórios
- 16 Mostrando conteúdo de diretório
- 17 Excluindo arquivos do SO
- 18 Copiando arquivos
- 19 Processando todos os arquivos de um diretório
- 20 Teste se Arquivo pode ser lido
- 21 Testar se Arquivo Permite Escrita
- 22 Testar se Arquivo Existe
- 23 Testar se é Arquivo ou Diretório
- 24 Outras Funções
- 25 Espaço Total no Disco
- 26 Espaço Livre no Disco
- 27 Tamanho de Diretório, número de arquivos e sub-diretórios
  - 27.1 Detalhes : $path
- 28 Tamanho de diretório
- 29 Trechos do Tutorial - The right way to read files with PHP da IBM
- 30 Lê e imprime todo o conteúdo de um arquivo CSV
- 31 rmdir -- Remove um diretório
- 32 rename -- Renomear um arquivo ($antigo, $novo)
- 33 unlink -- Apaga um arquivo
- 34 mkdir -- Criar um diretório
- 35 file_exists -- Checa se um arquivo ou diretório existe
- 36 disk_free_space -- Retorna o espaço disponivel no diretório
- 37 Receber Conteúdo de URL
- 38 Recursively find files by filename pattern
- 39 Referência

Abrir arquivo

fopen(filename, mode, [use_include_path]);

filename: pode ser simplesmente um nome, ou um caminho completo.

Exemplos: "arquivo.txt", "./arquivo.dat", "/data/data.txt".

mode: especifica o modo de abertura, ou seja, se o arquivo deve ser aberto para leitura, escrita, etc.
Modos de abertura:

```
- r: abre o arquivo no modo somente leitura e posiciona o ponteiro no início do
arquivo;
        o arquivo já deve existir;
- r+: abre o arquivo para leitura/escrita, posiciona o ponteiro no início do
arquivo;
- w: abre o arquivo no modo somente escrita; se o arquivo já existir, será
sobrescrito;
        senão, será criado um novo;
- w+: abre o arquivo para escrita/leitura; se o arquivo já existir, será
sobrescrito;
        senão, será criado um novo;
- a: abre o arquivo para anexar dados, posiciona o ponteiro no final do arquivo;
        se o arquivo não existir, será criado um novo;
- a+: abre o arquivo para anexo/leitura, posiciona o ponteiro no final do
arquivo;
        se o arquivo não existir, será criado um novo;

<?php
        $fp = fopen("./arquivo.dat", "r"); // $fp conterá o handle do arquivo
que abrimos
?>
<pre>


<h2>Fechar Arquivo</h2>

<pre>
fclose(handle_arquivo);

<?php
fclose($fp);
?>
```

Gravar em Arquivo

```
<?php
$fp = fopen("./dados.txt", "w");
fwrite($fp, "Olá mundo do PHP!"); // grava a string no arquivo. Se não existir
será criado
fclose($fp);
?>
```

Ler Arquivo

```
<?php

$fp = fopen("./dados.txt", "r");
$texto = fread($fp, 20); // lê 20 bytes do arquivo e armazena em $text
fclose($fp);
echo $texto;
?>
```

fgets()

Esta função é usada na leitura de strings de um arquivo. fgets() lê "length \u2013 1" bytes do arquivo. Se for encontrado o final da linha e o número de bytes especificados não tiver sido atingido, fgets() terminará a leitura no final da linha (ou no final do arquivo, se for o caso). Eis a sua sintaxe:

fgets(handle, length);

- handle: handle do arquivo de onde os dados serão lidos;

- length: tamanho em bytes do buffer de leitura;

Exemplo:

```
<?php
$fp = fopen("./dados.txt", "r");
$texto = fgets($fp, 3);
fclose($fp);
echo $texto;
?>
```

fgetc()

Esta função permite ler caractere por caractere de um arquivo. Seguem a sintaxe e um exemplo de utilização:

fgetc(handle);

- handle: handle do arquivo de onde os dados serão lidos;

Ler Todo um Arquivo

```
<?php
$fp = fopen("./dados.txt", "r");
while (!feof($fp)){
    $char .= fgetc($fp);
}
fclose($fp);
echo $char."<br><br>";
?>
```

file()

Esta função lê um arquivo completo, e armazena cada linha do arquivo como um elemento de um array. Depois de ler todo o conteúdo do arquivo, file() o fecha automaticamente, não sendo necessária uma chamada a fclose(); Vejamos a sintaxe:

Ler Arquivo Via URL

$fh = fopen("http://127.0.0.1/", "r");

file(filename);

- filename: nome ou caminho completo de um arquivo.

Exemplo:

```
<?php
// file() lê todo o arquivo
$file_lines = file("./dados.txt");
echo "Primeira linha: " . $file_lines[0]."<br>";
echo "Segunda linha: " . $file_lines[1]."<br>";
echo "Terceira linha: " . $file_lines[2];
?>
```

Além dessas funções para leitura e escrita, existe ainda uma função bastante útil, que testa se o final do arquivo foi atingido. É a função feof(), que tem a seguinte sintaxe:

feof(handle)

- handle: handle do arquivo;

Exemplo:

```
<?php
$fp = fopen("./dados.txt", "r");
while(!feof($fp)) {
    $char .= fgetc($fp);
}
fclose($fp);
echo $char;
?>
```

Contando o Número de Linhas de um arquivo

```
<?php
// Contar o número de linhas de um arquivo, iniciando com 1
$fp = "./dados.txt";
$line_count = count (file ($fp));
echo $line_count;
?>
```

Contar palavras de um arquivo mostrando duplicadas

```
/*
// Contar palavras repetidas em um arquivo
$fn = "./dados.txt";
$f_contents = preg_split ("/[\s+/", implode ("", file ($fn)));
foreach ($f_content as $palavra) {
        $ar[$palavra]++;
}
print "A seguinte palavra tem duplicatas<br>";
foreach ($ar as $palavra => $conta_palavra) {
        if (conta_palavra > 1) {
                print "Palavra: $palavra<br>Número de ocorrências:
$conta_palavra<br><br>";
        }
}
*/
```

Ler de forma inversa um arquivo, linha a linha

```
<?php
$fn = "./dados.txt";
$f_contents = array_reverse (file ($fn));
foreach ($f_contents as $linha_inversa) {
        print $linha_inversa;
}
?>
```

Ler aleatoriamente linha de arquivo

```
<?php
$fn = "./pensamentos.txt";
$f_contents = file ($fn);
srand ((double)microtime()*1000000);
$linha_aleatoria = $f_contents[ rand (0, (count ($f_contents) - 1)) ];
print $linha_aleatoria;
?>
```

Ler linha específica de arquivo

```
<?php
$fn = "./dados.txt";
$nr_linha = 38;
$f_contents = file ($fn);
$sua_linha = $f_contents [$nr_linha];
print $sua_linha;
?>
```

Operações com Diretórios

Mostrando conteúdo de diretório

```
<?php
$dn = opendir ("/home/1www/");
while ($file = readdir ($dn)) {
        print "$file<br>";
}
closedir($dn);
?>
```

Excluindo arquivos do SO

```
<?php
$fn = "./dados0.txt";
// Excluindo arquivo
$ret = unlink ($fn);
if ($ret){
         die ("Arquivo excluído!");
}else{
        die ("Erro ao excluir arquivo");
}
?>
```

Copiando arquivos

```
<?php
$fn = "./dados.txt";

if (copy ($fn, "dados0.txt")){
         die ("Arquivo '$fn' copiado para dados0.txt ");
}else{
        die ("Erro ao copiar arquivo");
}
?>
```

Processando todos os arquivos de um diretório

```
<?php
$dh = dir ("/home/1www/");
while ($entrada = $dh->read()) {
        print $entrada . "<br>";
}
$dh->close();
?>
```

Teste se Arquivo pode ser lido

```
<?php
if (is_readable('http://127.0.0.1/index.html')) {
   header('Location: http://127.0.0.1/index.html');
}else{
        echo "Este arquivo não pode ser lido!";
}

?>
```

```
<?php
// TESTAR SE ARQUIVO PERMITE LEITURA

print '<br>';
$filename = 'teste2.php';
if (is_readable($filename)) {
   echo 'O arquivo permite leitura';
} else {
   echo 'O arquivo não permite leitura';
}
print '<br>';
```

//Outro

<?php if (is_readable('http://127.0.0.1/index.html')) {

```
  header('Location: http://127.0.0.1/index.html');
```

}else{ echo "Este arquivo não pode ser lido!"; }

?>

Testar se Arquivo Permite Escrita

```
if (is_writable($filename)) {
   echo 'O arquivo permite escrita';
} else {
   echo 'O arquivo não permite escrita';
}

?>
```

Testar se Arquivo Existe

```
<?php
print '<br>';
// TESTAR SE ARQUIVO EXISTE
$filename = 'teste2.php';

if (file_exists($filename)) {
   echo "O arquivo $filename existe";
```

```
} else {
   echo "O arquivo $filename não existe";
}
?>
```

Testar se é Arquivo ou Diretório

```
<?php
print '<br>';
// TESTAR SE ARQUIVO É UM ARQUIVO COMUN OU SE É DIRETÓRIO
$filename = 'teste2.php';

$filename2 = 'c:\windows';

if (is_file($filename)) {
   echo "O arquivo $filename é comun";
}else{
  echo "O arquivo $filename não é um arquivo comun";
}
print '<br>';

if (is_file($filename2)){
   echo "O arquivo $filename2 é comun";
}else{
  echo "O arquivo $filename2 não é um arquivo comun";
}

print '<br>';

if (is_dir($filename2)){
   echo "$filename2 é um diretório";
}else{
  echo "$filename2 não é um diretório";
}
```

Outras Funções

```
is_link($diretorio)
readlink($dir_link) // retorna o path completo do link
bool symlink ( string $destino, string $linkorigem ) // Cria um link simbólico
```

Espaço Total no Disco

<?php // ESPAÇO TOTAL NO DISCO $diretorio="c:/"; print disk_total_space($diretorio); print "
";

Espaço Livre no Disco

// ESPAÇO LIVRE NO DISCO // $df contém o número de bytes disponível em "/" $df = disk_free_space("c:/"); print $df ?>

Tamanho de Diretório, número de arquivos e sub-diretórios

<?php // CALCULANDO TAMANHO OCUPADO POR UM DIRETÓRIO, NR DE ARQUIVOS E SUBDIRETÓRIOS // http://www.go4expert.com/forums/showthread.php?t=290

function getDirectorySize($path) {

```
 $totalsize = 0;
 $totalcount = 0;
 $dircount = 0;
 if ($handle = opendir ($path))
 {
   while (false !== ($file = readdir($handle)))
   {
     $nextpath = $path . '/' . $file;
     if ($file != '.' && $file != '..' && !is_link ($nextpath))
     {
       if (is_dir ($nextpath))
       {
         $dircount++;
         $result = getDirectorySize($nextpath);
         $totalsize += $result['size'];
         $totalcount += $result['count'];
         $dircount += $result['dircount'];
       }
       elseif (is_file ($nextpath))
       {
         $totalsize += filesize ($nextpath);
         $totalcount++;
       }
     }
   }
 }
 closedir ($handle);
 $total['size'] = $totalsize;
 $total['count'] = $totalcount;
 $total['dircount'] = $dircount;
 return $total;

}
```

function sizeFormat($size) {

```
   if($size<1024)
   {
       return $size." bytes";
   }
   else if($size<(1024*1024))
   {
       $size=round($size/1024,1);
```

```
        return $size." KB";
    }
    else if($size<(1024*1024*1024))
    {
        $size=round($size/(1024*1024),1);
        return $size." MB";
    }
    else
    {
        $size=round($size/(1024*1024*1024),1);
        return $size." GB";
    }

}
```

// Usando

$path="D:/_xampplite/htdocs/desweb/7AplicativosExemplo/extras"; $ar=getDirectorySize($path);

echo "

Detalhes : $path

";

echo "Tamanho total : ".sizeFormat($ar['size'])."
"; echo "No. de arquivos : ".$ar['count']."
"; echo "No. de diretórios : ".$ar['dircount']."
"; ?>

Tamanho de diretório

<? // CALCULANDO TAMANHO (BYTES) OCUPADO POR UM DIRETÓRIO // http://www.weberdev.com/get_example-4171.html

function dir_size($dir) {

```
    if( !$dir or !is_dir( $dir ) )
    {
        return 0;
    }

    $ret = 0;
    $sub = opendir( $dir );
    while( $file = readdir( $sub ) )
    {
        if( is_dir( $dir . '/' . $file ) && $file !== ".." && $file !== "." )
        {
            $ret += dir_size( $dir . '/' . $file );
            unset( $file );
        }
        elseif( !is_dir( $dir . '/' . $file ) )
        {
            $stats = stat( $dir . '/' . $file );
            $ret += $stats['size'];
```

```
            unset( $file );
        }
    }
    closedir( $sub );
    unset( $sub );
    return $ret;

}
```

echo dir_size("D:/_xampplite/htdocs/desweb/1LinguagemPHP/php/tutoriais/trabalhando_com"); ?>

<?php

// MAIS UMA ÓTIMA FUNÇÃO PARA LER O TAMANHO DE UM DIRETÓRIO

/*

```
* PHP Freaks Code Library
* http://www.phpfreaks.com/quickcode.php
*
* Title: Directory Size
* Version: 1.0
* Author: Nathan Taylor aka(Lakario)
* Date: Saturday, 12/20/2003 - 12:34 PM
*
*
*
* NOTICE: This code is available from PHPFreaks.com code Library.
*         This code is not Copyrighted by PHP Freaks.
*
*         PHP Freaks does not claim authorship of this code.
*
*         This code was submitted to our website by a user.
*
*         The user may or may not claim authorship of this code.
*
*         If there are any questions about the origin of this code,
*         please contact the person who submitted it, not PHPFreaks.com!
*
*         USE THIS CODE AT YOUR OWN RISK! NO GUARANTEES ARE GIVEN!
*
* SHAMELESS PLUG: Need WebHosting? Checkout WebHost Freaks:
*                 http://www.webhostfreaks.com
*                 WebHosting by PHP Freaks / The Web Freaks!
```

- /

// * Description / Example: // * // * This code will allow an individual to quickly obtain the size and number of files inside a directory recursively. // * // * It also includes a convenient byte value converter to kilobyte, megabyte, gigabyte, or trilobyte accordingly.

?>

<?php function DirStat($directory) { global $FolderCount, $FileCount, $FolderSize;

chdir($directory); $directory = getcwd(); if($open = opendir($directory)) { //while($file = readdir($open)) { while(false !== ($file = readdir($open))) { if($file == '..' || $file == '.') continue;

```
if(is_file($file)) { $FileCount++; $FolderSize += filesize($file); } elseif(is_dir($file))
{ $FolderCount++; } } if($FolderCount > 0) { $open2 = opendir($directory); while($folders =
readdir($open2)) { $folder = $directory.'/'.$folders; if($folders == '..' || $folders == '.') continue;
if(is_dir($folder)) { DirStat($folder); } } closedir($open2); } closedir($open); } }

function ByteSize($bytes) { $size = $bytes / 1024; if($size < 1024){ $size = number_format($size,
2); $size .= 'kb'; } else { if($size / 1024 < 1024) { $size = number_format($size / 1024, 2); $size .=
'mb'; } elseif($size / 1024 / 1024 < 1024) { $size = number_format($size / 1024 / 1024, 2); $size .=
'gb'; } else { $size = number_format($size / 1024 / 1024 / 1024, 2); $size .= 'tb'; } } return $size; }

$folder = 'D:/1Enviar/Hoje'; $dir = getcwd(); DirStat($folder, 0); chdir($dir); $FolderSize =
ByteSize($FolderSize);

echo 'Folder Name: '.$folder.'
'.chr(10); echo 'File Count: '.$FileCount.'
'.chr(10); echo 'Folder Size:'.$FolderSize.'
'.chr(10); ?>
```

Trechos do Tutorial - The right way to read files with PHP da IBM

USANDO fscanf

fscanf

Coming back to string processing, fscanf again follows the traditional C file library functions. If you're unfamiliar with it, fscanf reads field data into variables from a file.

```
list ($field1, $field2, $field3) = fscanf($fh, "%s %s %s");
```

FUNÇÃO fpassthru

No matter how you've been reading your file, you can dump the rest of your data to your standard output channel using fpassthru.

```
fpassthru($fh);
```

```
my_file = file_get_contents("myfilename");
echo $my_file;
```

Although it isn't best practice, you can write this command even more concisely as:

```
echo file_get_contents("myfilename");
```

This article is primarily about dealing with local files, but it's worth noting that you can grab, echo, and parse other Web pages with these functions, as well.

```
echo file_get_contents("http://127.0.0.1/");
```

This command is effectively the same as:

$fh = fopen("http://127.0.0.1/", "r");

fpassthru($fh);

You must be looking at this and thinking, "That's still way too much effort." The PHP developers agree with you. So you can shorten the above command to:

readfile("http://127.0.0.1/");

The readfile function dumps the entire contents of a file or Web page to the default output buffer. By default, this command prints an error message if it fails. To avoid this behavior (if you want to), try:

@readfile("http://127.0.0.1/");

Of course, if you actually want to parse your files, the single string that file_get_contents returns might be a bit overwhelming. Your first inclination might be to break it up a little bit with the split() function.

$array = split("\n", file_get_contents("myfile"));

But why go through all that trouble when there's a perfectly good function to do it for you? PHP's file() function does this in one step: It returns an array of strings broken up by lines.

$array = file("myfile");

It should be noted that there is a slight difference between the above two examples. While the split command drops the newlines, the newlines are still attached to the strings in the array when using the file command (as with the fgets command).

PHP's power goes far beyond this, though. You can parse entire PHP-style .ini files in a single command using parse_ini_file. The parse_ini_file command accepts files similar to Listing 4.

Listing 4. A sample .ini file

```
; Comment
[personal information]
name = "King Arthur"
quest = To seek the holy grail
favorite color = Blue

[more stuff]
Samuel Clemens = Mark Twain
Caryn Johnson = Whoopi Goldberg
```

The following commands would dump this file into an array, then print that array:

```
$file_array = parse_ini_file("holy_grail.ini");
print_r $file_array;
```

The following output is the result:

```
Listing 5. Output
Array
(
    [name] => King Arthur
    [quest] => To seek the Holy Grail
    [favorite color] => Blue
    [Samuel Clemens] => Mark Twain
    [Caryn Johnson] => Whoopi Goldberg
)
```

Of course, you might notice that this command merged the sections. This is the default behavior, but you can fix it easily by passing a second argument to parse_ini_file: process_sections, which is a Boolean variable. Set process_sections to True.

$file_array = parse_ini_file("holy_grail.ini", true);

print_r $file_array;

And you'll get the following output:

Listing 6. Output

```
Array
(
    [personal information] => Array
        (
            [name] => King Arthur
            [quest] => To seek the Holy Grail
            [favorite color] => Blue
        )

    [more stuff] => Array
        (
            [Samuel Clemens] => Mark Twain
            [Caryn Johnson] => Whoopi Goldberg
        )

)
```

PHP placed the data into an easily parsable multidimensional array.

This is just the tip of the iceberg when it comes to PHP file processing. More complex functions like tidy_parse_file and xml_parse can help you handle HTML and XML documents, respectively. See Resources for details on how these particular functions work. These are well worth looking at if you'll be dealing with those types of files, but instead of considering every possible file type you might run into in detail in this article, here are a few good general rules for dealing with the

functions I've described thus far.

Good practice

Never assume that everything in your program will work as planned. For example, what if the file you're looking for has moved? What if the permissions have been altered and you're unable to read the contents? You can check for these things in advance by using file_exists and is_readable.

Listing 7. Use file_exists and is_readable

```
$filename = "myfile";
if (file_exists($filename) && is_readable ($filename)) {
        $fh = fopen($filename, "r");
        # Processing
        fclose($fh);
}

In practice, however, such code is probably overkill. Processing the return
value of fopen is simpler and more accurate.

if ($fh = fopen($filename, "r")) {
        # Processing
        fclose($fh);
}
```

Final do trecho do tut IBM----------

Lê e imprime todo o conteúdo de um arquivo CSV

```
<?php
// Lê e imprime todo o conteúdo de um arquivo CSV
$row = 1;
$handle = fopen ("test.csv","r");
while ($data = fgetcsv ($handle, 1000, ",")) {
   $num = count ($data);
   print "<p> $num campos na linha $row: <br>";
   $row++;
   for ($c=0; $c < $num; $c++) {
       print $data[$c] . "<br>";
   }
}
fclose ($handle);
?>
<pre>

Outra:
<pre>
<?php

define('CSV_BOTH', 1);
define('CSV_ASSOC', 2);
define('CSV_NUM', 3);

function parse_csv($filename, $result_type = CSV_BOTH) {
```

```
    if(!file_exists($filename)) {
        die("file (" . $filename . ") does not exist\n");
    }

    $lines = file($filename);

    $title_line = trim(array_shift($lines));
    $titles = split(",", $title_line);

    $records = array();
    foreach($lines as $line_num => $line) {
        $subject = trim($line);
        $fields = array();
        for($field_num = 0; $field_num < count($titles); $field_num++) {
            if($subject{0} == '"') {
                preg_match('/^"(([^"]|\\")*)",?(.*)$/', $subject, $matches);

                $value = $matches[1];
                $subject = $matches[3];

                if($result_type == CSV_BOTH || $result_type == CSV_ASSOC) {
                    $fields[$titles[$field_num]] = $value;
                }

                if($result_type == CSV_BOTH || $result_type == CSV_NUM) {
                    $fields[$field_num] = $value;
                }
            } else {
                preg_match('/^([^,]*),?(.*)$/', $subject, $matches);

                $value = $matches[1];
                $subject = $matches[2];

                if($result_type == CSV_BOTH || $result_type == CSV_ASSOC) {
                    $fields[$titles[$field_num]] = $value;
                }

                if($result_type == CSV_BOTH || $result_type == CSV_NUM) {
                    $fields[$field_num] = $value;
                }
            }
        }

        $records[] = $fields;
    }

    return $records;
}

?>
<pre>

This version is conditional - it only adds quotes if needed:
<pre>
<?
function csv_escape($str) {
    $str = str_replace(array('"', ',', "\n", "\r"), array('""', ',', "\n", "\r"),
$str, &$count);
    if($count) {
```

```
        return '"' . $str . '"';
    } else {
        return $str;
    }
}
?>
```

```
<?php
$caminho = "/home/httpd/html/index.php";
$arquivo = basename ($caminho);        // $arquivo = "index.php"
$arquivo = basename ($caminho,".php"); // $arquivo = "index"
?>
```

rmdir -- Remove um diretório

This functions deletes or empties the directory. Without using recursive functions!

```
<?php

/**
 * Removes the directory and all its contents.
 *
 * @param string the directory name to remove
 * @param boolean whether to just empty the given directory, without deleting
the given directory.
 * @return boolean True/False whether the directory was deleted.
 */
function deleteDirectory($dirname,$only_empty=false) {
    if (!is_dir($dirname))
        return false;
    $dscan = array(realpath($dirname));
    $darr = array();
    while (!empty($dscan)) {
        $dcur = array_pop($dscan);
        $darr[] = $dcur;
        if ($d=opendir($dcur)) {
            while ($f=readdir($d)) {
                if ($f=='.' || $f=='..')
                    continue;
                $f=$dcur.'/'.$f;
                if (is_dir($f))
                    $dscan[] = $f;
                else
                    unlink($f);
            }
            closedir($d);
        }
    }
    $i_until = ($only_empty)? 1 : 0;
    for ($i=count($darr)-1; $i>=$i_until; $i--) {
        echo "\nDeleting '".$darr[$i]."' ... ";
        if (rmdir($darr[$i]))
            echo "ok";
        else
            echo "FAIL";
```

```
   }
   return (($only_empty)? (count(scandir)<=2) : (!is_dir($dirname)));
}

?>

//Outra

<?php
/* Function to remove directories, even if they contain files or
subdirectories.  Returns array of removed/deleted items, or false if nothing
was removed/deleted.

by Justin Frim.  2007-01-18

Feel free to use this in your own code.
*/

function rmdirtree($dirname) {
   if (is_dir($dirname)) {    //Operate on dirs only
       $result=array();
       if (substr($dirname,-1)!='/') {$dirname.='/';}    //Append slash if
necessary
       $handle = opendir($dirname);
       while (false !== ($file = readdir($handle))) {
           if ($file!='.' && $file!= '..') {    //Ignore . and ..
               $path = $dirname.$file;
               if (is_dir($path)) {    //Recurse if subdir, Delete if file
                   $result=array_merge($result,rmdirtree($path));
               }else{
                   unlink($path);
                   $result[].=$path;
               }
           }
       }
       closedir($handle);
       rmdir($dirname);    //Remove dir
       $result[].=$dirname;
       return $result;    //Return array of deleted items
   }else{
       return false;    //Return false if attempting to operate on a file
   }
}
?>
```

rename -- Renomear um arquivo ($antigo, $novo)

```
<?php
  rename("/tmp/tmp_file.txt", "/home/user/login/docs/my_file.txt");
?>
```

unlink -- Apaga um arquivo

mkdir -- Criar um diretório

mkdir ("/path/to/my/dir", 0700);

is_file -- Diz se o arquivo é um arquivo comum (não é diretório)

file -- Le um arquivo inteiro para um array

```
<?php
// Le um arquivo em um array. Nesse exemplo você pode obter via HTTP para obter
// o código fonte HTML de uma URL.
$lines = file ('http://www.exemplo.com/');

// Roda através do array, mostrando o fonte HTML com numeração de linhas.
foreach ($lines as $line_num => $line) {
   echo "Linha #<b>{$line_num}</b> : " . htmlspecialchars($line) . "<br>\n";
}

// Outro exemplo, onde obtemos a página web inteira como uma string. Veja também
file_get_contents().
$html = implode ('', file ('http://www.exemplo.com/'));
?>
```

file_exists -- Checa se um arquivo ou diretório existe

```
$filename = '/caminho/para/qualquer.txt';

if (file_exists($filename)) {
   print "O arquivo $filename existe";
} else {
   print "O arquivo $filename não existe";
}
```

disk_free_space -- Retorna o espaço disponivel no diretório

Receber Conteúdo de URL

<?php // Trazer conteúdo de arquivo ou de página para string

// Define a context for HTTP. $aContext = array(

```
  'http' => array(
      'proxy' => 'tcp://10.0.0.1:3128', // This needs to be the server and the
port of the NTLM Authentication Proxy Server.
      'request_fulluri' => True,
      ),
```

```
  );
```

$cxContext = stream_context_create($aContext);

// Now all file stream functions can use this context. $sFile = file_get_contents("http://www.google.com", False, $cxContext); echo $sFile; ?>

Recursively find files by filename pattern

http://snippets.dzone.com/posts/show/4147

Scans a directory, and all subdirectories for files, matching a regular expression. Each match is sent to the callback provided as third argument. A simple example:

function my_handler($filename) {

```
 echo $filename . "\n";
```

} find_files('c:/', '/php$/', 'my_handler');

And the actual snippet

function find_files($path, $pattern, $callback) {

```
 $path = rtrim(str_replace("\\", "/", $path), '/') . '/';
 $matches = Array();
 $entries = Array();
 $dir = dir($path);
 while (false !== ($entry = $dir->read())) {
   $entries[] = $entry;
 }
 $dir->close();
 foreach ($entries as $entry) {
   $fullname = $path . $entry;
   if ($entry != '.' && $entry != '..' && is_dir($fullname)) {
     find_files($fullname, $pattern, $callback);
   } else if (is_file($fullname) && preg_match($pattern, $entry)) {
     call_user_func($callback, $fullname);
   }
 }
```

}

Referência

http://phpbrasil.com/articles/print.php/id/310

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Arquivos"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Path

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 PATH
- 2 Exemplos simples de uso de funções para path do PHP
- 3 Recebendo o Path Absoluto do Script Atual
- 4 Recebendo o path relativo do webserver do script atual

PATH

Exemplos simples de uso de funções para path do PHP

```
<?php
   $path=dirname(realpath($_SERVER['SCRIPT_FILENAME']));
   $path=substr($path,0,strlen($path) - 5);
echo "<br>Path deste script sem 5 finais caracteres - " . $path;

echo "<br><br>Diretório atual - ".dirname(__FILE__);
echo "<br>Caminho completo do script atual - ".__FILE__;

echo "<br>URL do script atual - " . "http://" . $_SERVER['HTTP_HOST'] .
$HTTP_SERVER_VARS["SCRIPT_NAME"];
?>

<?php
$path = "/etc/passwd";
$file = dirname($path); // $file is set to "/etc"
?>

<?php
$path = "/home/httpd/html/index.php";
$file = basename($path);        // $file is set to "index.php"
$file = basename($path, ".php"); // $file is set to "index"
?>

<?php
echo dirname($_SERVER["REQUEST_URI"]);
?>
```

Recebendo o Path Absoluto do Script Atual

dirname(__FILE__)

Recebendo o path relativo do webserver do script atual

```
function GetRelativePath($path)
{
   $npath = str_replace('\\', '/', $path);
   return str_replace(GetVar('DOCUMENT_ROOT'), '', $npath);
}
```

GetRelativePath(dirname(__FILE__));

<?php

if (DIRECTORY_SEPARATOR=='/')

```
 $absolute_path = dirname(__FILE__).'/';
```

else

```
 $absolute_path = str_replace('\\\\', '/', dirname(__FILE__)).'/';
```

?>

Resultará em um path absoluto no estilo UNIX que funciona também em PHP5 sob Windows.

Em algumas instalações (< 4.4.1) $_SERVER['REQUEST_URI'] não está configurado, usado o código para corrigir:

```
<?php

if (!isset($_SERVER['REQUEST_URI'])) {
   $_SERVER['REQUEST_URI'] = substr($_SERVER['PHP_SELF'],1);
   if (isset($_SERVER['QUERY_STRING']))
$_SERVER['REQUEST_URI'].='?'.$_SERVER['QUERY_STRING'];
}

?>

$my_uri = "http://" . $_SERVER['HTTP_HOST'] . $HTTP_SERVER_VARS["SCRIPT_NAME"];

// então

<?php echo ("$my_uri");?>

<?php
include("{$_SERVER['DOCUMENT_ROOT']}/includes/my_include.php");
?>
```

Você pode usar isso para receber o diretório pai:

dirname(dirname(__FILE__))

...include a file relative to file path:

include(dirname(__FILE__) . '/path/relative/file_to_include.php');

Isso colocará ambos os paths "www" e "file" de forma fácil para transportar o array.

```
<?php
// build the www path:
$me = $_SERVER['PHP_SELF'];
$Apathweb = explode("/", $me);
$myFileName = array_pop($Apathweb);
$pathweb = implode("/", $Apathweb);
$myURL = "http://".$_SERVER['HTTP_HOST'].$pathweb."/".$myFileName;
$PAGE_BASE['www'] = $myURL;

// build the file path:
strstr( PHP_OS, "WIN") ? $strPathSeparator = "\\" : $strPathSeparator = "/";
$pathfile = getcwd ();
$PAGE_BASE['physical'] = $pathfile.$strPathSeparator.$myFileName;

// this is so you can verify the results:
$www = $PAGE_BASE['www'];
$physical = $PAGE_BASE['physical'];

echo "$physical<p>";
echo "$www<p>";
?>
```

retornará algo como:

Windows:

F:\dev\Inetpub\wwwroot\somedirectory\index.php

http://devserver/somedirectory/index.php

Unix:

/home/somepathto/gieson.com/webroot/index.php

http://www.gieson.com/index.php

Path absoluto do script em execução

$path=dirname(realpath($_SERVER['SCRIPT_FILENAME']));

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Path"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Includes

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Includes

Inclue e avalia o conteúdo do arquivo.

Require

Também inclue e avalia o conteúdo do arquivo incluído.

A diferença entre ambos é que o include ao encontrar um erro, lança um warning apenas, enquanto que o require lança um Fatal Error, que pára o processamento.

Uso do require: para códigos que requerem maior segurança.

Include_Once e Require_Once

São semelhantes ao include e require, sendo que estes incluem um arquivo somente uma vez.

Mostrando uso do require_once:

echo.php

```
<?php
echo "Hello";
?>
```

teste.php

```
<?php
require('echo.php');
require_once('echo.php');
?>
```

Chamar no navegador o script teste.php

saída: "Hello".

Agora teste2.php:

```
<?php
require('echo.php');
require('echo.php');
?>
```

Executar teste2.php

saída: "HelloHello".

Agora teste3.php:

```
<?php
require_once('echo.php');
require('echo.php');
?>
```

Executar teste3.php

saída: "HelloHello".

Ou seja, ao encontrar require_once, ele verifica se o arquivo já foi incluído, e somente o incluirá novamente se ele ainda não tiver sido incluído.

```
<?php
// Isto está errado e não funcionará como desejado
if ($condition)
   include $arquivo;
else
   include $outro;

// E este está correto
if ($condition) {
   include $arquivo;
} else {
   include $outro;
}

?>

<?php
$path="/full/path/to/script/";
if (getdomain($path) == 'yourdomain'){
     include($path.'somefile.php');
}
?>
```

"variables.php"

```
<?php
$includer = basename($_SERVER['SCRIPT_NAME']);

switch ($includer) {
   case 'a.php':
   $this_variable = 'included by script a.php';
   break;
```

```
   case 'b.php':
   $this_variable = 'included by script b.php';
   break;

   default:
   $this_variable = 'included by unkown script';
}
echo $this_variable;
?>
```

Test with 3 different files "a.php", "b.php", "c.php", all with the same content:

```
<?php
include 'variables.php';
?>
```

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Includes"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Formatação da saída

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 printf -- Mostra uma string formatada
- 2 sscanf
- 3 fscanf
- 4 sprintf
- 5 vprintf

printf -- Mostra uma string formatada

void printf (string format [, mixed args])

sscanf

Interpreta a entrada de uma string de acordo com um formato mixed sscanf (string str, string formato [, string var1])

```
<?php
// Pegando o número serial
$serial = sscanf("SN/2350001","SN/%d");
// e a data de criação
$mandate = "January 01 2000";
list($month, $day, $year) = sscanf($mandate,"%s %d %d");
echo "O Item $serial foi criado em: $year-".substr($month,0,3)."-$day\n";
?>
```

Se parâetros opcionais são passados, a função retornará o número de valores assumidos. Os parâmetros opcionais devem ser passados por referência.

Exemplo 2. sscanf() - usando parâmetros opcionais

```
<?php
// pega informação do autor e gera uma entrada de DocBook
$auth = "24\tLewis Carroll";
$n = sscanf($auth,"%d\t%s %s", &$id, &$first, &$last);
```

```
echo "<author id='$id'>
   <firstname>$first</firstname>
   <surname>$last</surname>
</author>\n";
?>
```

fscanf

Interpreta a leitura de um arquivo de acordo com um formato

mixed fscanf (resource handle, string formato [, string var1])

```
$handle = fopen ("users.txt","r");
while ($userinfo = fscanf ($handle, "%s\t%s\t%s\n")) {
   list ($name, $profission, $countrycode) = $userinfo;
   //... fazer algo com os valores
}
fclose($handle);

$goodevil = array ('There is a difference between %s and %s', 'good', 'evil');
echo call_user_func_array('sprintf',$goodevil);

<?php
$heading1 = "Label 1";
$heading2 = "Label 2";

$value1 = "31298";
$value2 = "98";

print "<pre>\n";
printf ("%'.-15.15s%'.6.6s\n", $heading1, $value1);
printf ("%'.-15.15s%'.6.6s\n", $heading2, $value2);
print "
```

\n";

?>

<?php $f='<?php $f=%c%s%c; printf($f,39,$f,39); ?>'; printf($f,39,$f,39); ?>

sprintf

Retorna uma string formatada

string sprintf (string format [, mixed args])

Um especificador de tipo que diz que o argumento deve ser tratado como do tipo. Os tipos possivéis são:

% - Um caractere porcento. Não é requerido neenhum argumento. b - O argumento é tratado com um inteiro, e mostrado como um binário. c - O argumento é tratado como um inteiro, e mostrado como o caractere ASCII correspondente. d - O argumento é tratado como um inteiro, e mostrado como um número decimal com sinal. u - O argumento é tratado com um inteiro, e mostrado como

um número decimal sem sinal. f - O argumento é tratado como um float, e mostrado como um número de ponto flutuante. o - O argumento é tratado com um inteiro, e mostrado como un número octal. s - O argumento é tratado e mostrado como uma string. x - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras minúsculas). X - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras maiúsculas).

```
<?php
$format = "There are %d monkeys in the %s";
printf($format,$num,$location);
?>
```

Este deve mostrar, "There are 5 monkeys in the tree". Mas imagine que nós estejamos criando a string de formatação em um arquivo separado, normalmente para internacionalizar e rescrevemos como:

Exemplo 2. Troca de argumentos

```
<?php
$format = "The %s contains %d monkeys";
printf($format,$num,$location);
?>
```

Agora nós temos um problema. A ordem dos argumentos na string de formatação não combina com os argumentos no código. Nós gostariamos de deixar o código como esta e simplesmente indicar na string de formatação quais argumentos pertencem aonde. Podemos escrever a string de formatação assim:

Exemplo 3. Troca de argumento

```
<?php
$format = "The %2\$s contains %1\$d monkeys";
printf($format,$num,$location);
?>
```

Um beneficio adicional disto é ue você pode repetir os especificadores de conversão sem adicionar mais argumentos em seu código. Por exemplo:

Exemplo 4. Troca de argumento

```
<?php
$format = "The %2\$s contains %1\$d monkeys.
           That's a nice %2\$s full of %1\$d monkeys.";
printf($format, $num, $location);
?>
```

Veja também printf(), sscanf(), fscanf(), vsprintf() e number_format().

Exemplos

Exemplo 5. sprintf(): inteiros prenchidos com zero

```
<?php
$isodate = sprintf("%04d-%02d-%02d", $year, $month, $day);
?>
```

Exemplo 6. sprintf(): formatando dinheiro

```
<?php
$money1 = 68.75;
$money2 = 54.35;
$money = $money1 + $money2;
// echo $money irá mostrar "123.1";
$formatted = sprintf("%01.2f", $money);
// echo $formatted irá mostrar "123.10"
?>
```

vprintf

Mostra uma string formatada

void vprintf (string formato, array args)

Mostra uma string formatada de acordo com o formato (o qual é descrito na documentação para a função sprintf()).

Funciona como printf() mas aceita uma matriz de argumentos, ao invés de um número variavel de argumentos.

```
<?php
$fruits = array(1, 'banana',1, 'apples', 3, 'oranges', 2, 'peaches');

vprintf("I have %d %s, %d %s, %d %s and %d %s.", $fruits);
?>
```

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Formata%C3%A7%C3%A3o_da_sa%C3%ADda"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Imagens e Gráficos

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 Trabalhando com a biblioteca gráfica GD
- 2 Gerando Thumbnails com GD
- 3 Gerando Imagens Dinamicamente
- 4 Desenhando retângulos
- 5 Desenhando polígonos
- 6 Desenhando arcos
- 7 Gerando Gráficos em PHP com a Biblioteca JPGraph
    - 7.1 O que é a JpGrapf?
    - 7.2 Requisitos
    - 7.3 Parâmetros de Compilação
- 8 Referência

Trabalhando com a biblioteca gráfica GD

(Se no Windows remover o ponto-e-vírgula ";" da linha "extension=php_gd.dll" do php.ini)

Gerando Thumbnails com GD

Artigo do BOZO no PHPBrasil:

http://phpbrasil.com/articles/article.php/id/1350

Gerando Imagens Dinamicamente

por Luiz Ribeiro

O PHP oferece uma interface ao módulo GD de Thomas Boutell. Usando tal módulo, você pode criar e editar imagens nos formatos JPEG e PNG. O formato GIF já foi aceito, mas como o algoritmo de compressão do GIF (LZW) contém uma patente de posse da Unisys, os desenvolvedores do módulo foram obrigados à retirar o suporte a esse formato nas versões mais recentes.

Bom, para iniciar vou explicar o procedimento para criar uma imagem usando o módulo GD em PHP. Se você não tem esse módulo, você pode fazer o download dele em http://www.boutell.com/gd/. Normalmente a GD acompanha uma instalação completa do PHP.

Para se criar a imagem, será usada a função ImageCreate(), então serão realizadas as alterações na imagem, então será finalizada a imagem usando ImageJpeg(), ImagePng() ou até ImageGif() se a versão do módulo GD for inferior à 1.4.

Bom, vamos ao que interessa. Primeiramente vamos criar uma pequena imagem com o seguinte texto: PHPBrasil. O código ficará da seguinte forma:

```
<?php
header("Content-type: image/gif"); //Informa ao browser que o arquivo é uma
imagem no formato GIF

$imagem = ImageCreate(150,40); //Cria uma imagem com as dimensões 100x20

$vermelho = ImageColorAllocate($imagem, 255, 0, 0); //Cria o segundo plano da
imagem e o configura para vermelho
$branco = ImageColorAllocate($imagem, 255, 255, 255); //Cria a cor de primeiro
plano da imagem e configura-a para branco

ImageString($imagem, 3, 3, 3, "PHPBrasil", $branco); //Imprime na imagem o texto
PHPBrasil na cor branca que está na variável $branco

ImageGif($imagem); //Converte a imagem para um GIF e a envia para o browser

ImageDestroy($imagem); //Destrói a memória alocada para a construção da imagem
GIF.
?>
```

Bom, o script está todo comentado e acho que você entendeu. Se alguma dúvida ficar martelando aí, manda um comentário. =D

Bom, neste exemplo usamos a função ImageGif() para converter a imagem, $imagem, e depoisa enviamos ao navegador. Mas poderíamos ter salvo esta imagem em um arquivo, ao invés de mostrar ela no navegador. Veja o exemplo:

```
<?php
$arquivo = "imagem1.gif";

$imagem = ImageCreate(150,40);

$vermelho = ImageColorAllocate($imagem, 255, 0, 0);
$branco = ImageColorAllocate($imagem, 255, 255, 255);

ImageString($imagem, 3, 3, 3, "PHPBrasil", $branco);
ImageGif($imagem, $arquivo);

ImageDestroy($imagem);

echo "A imagem foi salva no arquivo $arquivo.";
?>
```

Como você deve ter notado, apenas retiramos aquele header() (que informava ao browser que o arquivo era uma imagem), afinal este exemplo não irá mostrar a imagem no navegador e sim gravar ela em $arquivo, e também mudamos os parâmetros da função ImageGif() para salvar a imagem no arquivo.

Nesta parte do artigo, irei explicar como desenhar retângulos, polígonos e arcos.

Desenhando retângulos

Vamos ao primeiro exemplo, que irá desenhar um simples retângulo preenchido usando GD (o formato da imagem a seguir é PNG).

```
<?php
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
ImageFilledRectangle($imagem, 5, 10, 60, 14, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Bom, neste exemplo só há uma função nova, a função ImageFilledRectangle() que como seu próprio nome diz é uma função que cria um retângulo com as dimensões e posição informadas, e na cor azul, que foi definida na variável $azul.

Já para criar um retângulo sem preenchimento você simplesmente irá trocar a função ImageFilledRectangle() por ImageRectangle(). O exemplo ficará da seguinte forma:

```
<?php
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
```

```
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
ImageRectangle($imagem, 5, 10, 60, 14, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Como foi dito, este exemplo irá criar uma imagem com um retângulo sem preenchimento, mas sua borda terá a cor $azul.

Desenhando polígonos

Para desenhar polígonos, vamos usar a função ImagePolygon(), que irá criar um polígono sem preenchimento, e a função ImageFilledPolygon() que irá desenhar um polígono com preenchimento.

Em nosso primeiro exemplo, vamos desenhar um polígono com vértices de (12, 10), (15, 20), (50, 17) e (70, 10) com uma borda de azul-claro:

```
<?php
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
$pontos = array(12, 10, 15, 20, 50, 17, 70, 10);
ImagePolygon($imagem, $pontos, 4, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Bom, agora vamos criar um polígono preenchido, você já deve ter pensado que o código será o mesmo, mas ao invés de ImagePolygon() usaremos ImageFilledPolygon(), se você énsou isso, acertou em cheio. Vamos ver como ficaria nossa imagem com um retângulo preenchido:

```
<?php
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
$pontos = array(12, 10, 15, 20, 50, 17, 70, 10);
ImageFilledPolygon($imagem, $pontos, 4, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Desenhando arcos

Bom, agora vamos desenhar alguns arcos em nossas imagens, para isso vamos usar a função ImageArc(). Antes de começarmos, vou passar a sintaxe da função:

int ImageArc(int im, int cx, int cy, int w, int h, int s, int e, int co1);

Esta função desenha um arco em uma imagem, im, com uma posição inicial de X de cx e uma

posição inicial Y de cy. O arco é de largura w e altura h, com um ângulo inicial de s e um ângulo final de e, tudo na cor co1.

Agora que já entendemos a função ImageArc() vamos ao nosso primeiro exemplo que irá desenhar uma elipse:

```
<?php
header("Content-type: image/gif");
$imagem = ImageCreate(500, 100);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
ImageColorTransparent($imagem, $branco);
$vermelho = ImageColorAllocate($imagem, 20, 93, 233);
ImageArc($imagem, 40, 50, 50, 40, 0, 360, $vermelho);
ImageGif($imagem);
ImageDestroy($imagem);
?>
```

O código acima funciona, pois para ter uma elipse, você precisa de uma diferença de 360 graus entre a posição inicial e a posição final. Aplicando esse conhecimento, também podemos desenhar um círculo preenchido usando a função ImageFillToBorder(). (Note que isso é um círculo, não uma elipse, porque os parâmetros de largura e altura têm o mesmo valor.)

```
<?php
header("Content-type: image/gif");
$imagem = ImageCreate(500, 100);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
ImageColorTransparent($imagem, $branco);
$vermelho = ImageColorAllocate($imagem, 20, 93, 233);
ImageArc($imagem, 40, 50, 50, 50, 0, 360, $vermelho);
ImageFillToBorder($imagem, 50, 40, $vermelho);
ImageGif($imagem);
ImageDestroy($imagem);
?>
```

Observação: os exemplos acima foram retirados do livro PHP: Guia do Desenvolvedor que está na lista de livros recomendados da PHPBrasil. Nos exemplos só foram alterados os nomes de algumas variáveis para facilitar a compreensão.

Bom, esse é o básico do módulo GD. Você com certeza tem muito a explorar ainda, em breve vou trazer mais alguns artigos sobre o assunto, para os que se interessaram, ou não entenderam alguma função podem ver no manual do PHP todas as funções de imagem:
http://br.php.net/manual/pt_BR/ref.image.php

Gerando Gráficos em PHP com a Biblioteca JPGraph

O que é a JpGrapf?

A JpGraph é uma biblioteca orientada a objetos de criação de gráficos, inteiramente escrita em PHP, que tem como base a extensão GD2 ou GD1 que acompanha o PHP.

Pode ser utilizada para criar diversos tipos de gráficos, de maneira fácil e escrevendo um mínimo de código. Quando aliada a bancos de dados torna os gráficos ainda mais interessantes.

Requisitos

Apache (http://httpd.apache.org)

PHP (www.php.net)

JpGraph (http://www.aditus.nu/jpgraph/)

Parâmetros de Compilação

- Compilar o PHP com suporte a GD e às fontes TTF (Linux):

Sugestão da documentação oficial da JpGrapf:

```
./configure --prefix=/usr/share \
--datadir=/usr/share/php \
--with-apxs=/usr/sbin/apxs \
--libdir=/usr/share \
--includedir=/usr/include \
--bindir=/usr/bin \
--with-config-file-path=/etc \
--enable-mbstring --enable-mbregex \
--with-pdflib=/usr \
--with-mysql  \
--with-ttf-dir=/usr/lib \
--with-freetype-dir=/usr/lib \
--with-gd --enable-gd-imgstrttf --enable-gd-native-ttf \
--with-zlib-dir=/usr/lib \
--with-png-dir=/usr/lib --with-jpeg-dir=/usr/lib --with-xpm-dir=/usr/X11R6 \
--with-tiff-dir=/usr/lib \
--enable-ftp \
--enable-memory-limit --enable-safe-mode \
--bindir=/usr/bin \
--enable-bcmath -enable-calendar \
--enable-ctype --with-ftp \
--enable-magic-quotes \
--enable-inline-optimization \
--with-bz2 \
--with-iconv
```

- No Windows basta descomentar no php.ini o suporte à GD2.

Obs.: No código de um gráfico não pode haver nenhuma saída em HTML ou texto.

Captura do site oficial.

Versões

Para PHP4 é indicada a versão 1.19 da JpGrapf e para a versão 5 do PHP é indicada a versão 2.0 beta.

Download das Fontes TTF (Linux)

http://corefonts.sourceforge.net/

http://www.gnome.org/fonts/

Como saber se o PHP já tem o suporte à JpGrapf?

Executar a função phpinfo(), que deve retornar:

- GD support - enabled
- FreeTypeSupport – enabled
- JPG support – enabled
- PNG support – enabled
- WBMP support – enabled

Exemplos que podem ser encontrados no site official em

http://www.aditus.nu/jpgraph/pdf/jpgraphddda.pdf

Exemplos de Gráficos

Gráfico dos Poços perfurados pelo DNOCS de 1900 a 1999, por estado:

Este gráfico acessa um banco PostgreSQL. Num pequeno form alguém informa um ano entre 1900 e 1999 e recebe o gráfico correspondente. Caso não informe o ano retornará o gráfico de todos os anos.

Código abaixo:

```
//Pequneo form de consulta
<h2 align=center>Consulta de Poços - DNOCS</h2>
<form name=frmPocos action="barras_pocos.php" method="post">
<center>Que ano que deseja Consultar? (Todos = vazio)<input name="ano" size=10>
<input type=submit value=Consultar></center>
</form>

//Arquivo barras_pocos.php
<?php
// Inclusão da biblioteca
include ("jpgraph.php");
include ("jpgraph_bar.php");
// Conexão ao banco PostgreSQL e consulta
$db = pg_connect("host=10.0.0.100 dbname=banco port=5432 user=user
password=pass") or die(pg_last_error());
$ano=$_POST['ano'];
// Se não for informado o ano da pesquisa no form anterior, exibirá todos os
poços, caso contrário mostra
// somente os poços do ano solicitado
if ($ano == "")
$sql=pg_query($db,"SELECT estado, count(poco) as quant from
recursos_hidricos.pocos group by estado");
else
$sql=pg_query($db,"SELECT estado, count(poco) as quant from
recursos_hidricos.pocos where recursos_hidricos.pocos.ano = $ano group by
estado");

while($row = pg_fetch_array($sql)) {
$quant[] = $row[1];     //Este array ($quant[]) sera usado em um dos eixos
$estado[] = $row[0];   // Este em outro eixo
```

```
}
// Construção da base do gráfico
$graph = new Graph(650,350,"auto");
$graph->SetScale("textint"); //Exibir as escalas
$graph->img->SetMargin(50,50,70,50);  //Margens dos 4 lados
$graph->title->Set('DNOCS - Poços X Estado do ano de '.$ano); // Título do
gráfico
$graph->title->SetFont(FF_VERDANA, FS_BOLD, 16);//Fonte do título
$graph->AdjBackgroundImage(0.4,0.7,-1); //Tipo de background
$graph->xaxis->title->Set('Estados'); //Título do eixo X
$graph->xaxis->SetLabelAngle(30); //Ângulo dos labels do eixo X
$graph->xaxis->SetTickLabels('Estados');
$graph->xaxis->SetFont(FF_VERDANA, FS_BOLD);//Fonte para o título do eixo X
$graph->xaxis->SetTickLabels($estado);  // Recebe o array dos estados do banco
$graph->yaxis->title->Set('Poços');
$graph->yaxis->SetFont(FF_FONT1, FS_BOLD);
$graph->yaxis->title->SetFont(FF_FONT1, FS_BOLD);
$graph->SetShadow(); //Adicionar sombra ao gráfico

//Adicionar um tipo de gráfico (barras)
$bplot = new BarPlot($quant);  //Recebe o outro array do banco
$bplot->SetFillColor("lightgreen"); // Cor do gráfico
$bplot->value->Show();
$bplot->value->SetFont(FF_ARIAL,FS_BOLD); //Fonte
$bplot->value->SetAngle(45); //Ângulo
$bplot->value->SetColor("black","navy"); //Cores

$graph->Add($bplot); //Adicionar o gráfico à base
$graph->Stroke();
?>
```

Ao baixar a JpGraph e descompactar no diretório web, veja a documentação, que exibe inúmeros tipos de gráficos com seus respectivos códigos ao lado, como também o subdiretório samples que tem 337 exemplos de gráficos.

Referência

http://www.phpbrasil.com/articles/print.php/id/164

http://www.zend.com/zend/tut/tutsweat3.php (ótimo tutorial, em inglês)

http://phpbrasil.com/articles/print.php/id/315 (outro muito bom e em português)

http://www.phpfreaks.com/print.php?cmd=tutorial&tut_id=115 (este abordando uso do MySQL)

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Imagens_e_Gr%C3%A1ficos"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Números

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Trabalhando com Números em PHP

Muito Cuidado ao Lidar com Números em Ponto Flutuante

Teste em PHP

```
<?php
echo (int) ((0.1 + 0.7 ) * 10);
?>
```

Agora teste isso:

echo (int) ((0.2 + 0.7) * 10);

Não conclua muito apressadamente que é deficiência do PHP.

Neste momento devemos ter conhecimento de como se comportam os números, especialmente os floats, que são normalizados pelo IEEE.

Teste em Java

```
class teste {
    public static void main(String[] args) {
        System.out.println((int) ((0.1 + 0.7 ) * 10)); //Display the string.
    }
}
```

Em Java também dá o mesmo resultado do PHP, o que leva a crer que a coisa não depende da linguagem mas das normas de como foram construídos os números pelo IEEE.

O Effective Java sugere que se use int, long ou BigDecimal para representar os valores monetários. A classe BigDecimal foi desenvolvida para resolver dois tipos de problemas associados a números de ponto flutuante (floats e doubles): primeiro, resolve o problema da inexatidão da representação de números decimais; segundo, pode ser usado para trabalhar com números com mais de 16 dígitos significativos. Em compensação, utilizar BigDecimal pode tornar o programa menos legível por não haver sobrecarga dos operadores matemáticos para ela, sendo necessário usar métodos da classe. Veja, por exemplo, como você faria o programa da listagem 1 com BigDecimal:

BigDecimal d1 = new BigDecimal("1.95");

BigDecimal d2 = new BigDecimal("1.03");

System.out.println(d1.subtract(d2));

Utilizar os primitivos normalmente é mais rápido e mais prático, mas o problema fica por conta da definição das casas decimais. Você pode controlar diretamente as casas decimais, por exemplo, utilizando como unidade para os valores o centavo ao invés de real. Um int ou um long passariam a representar a quantidade de centavos presentes no valor, e não a quatidade de reais. Por exemplo:

long l1 = 195;

long l2 = 103;

System.out.println(l1 ? l2);

Listagem 6: Programa da listagem 1 com long

As variáveis acima dizem que você tem 195 centavos (e não R$ 1,95) e vai gastar 103 centavos, e não R$ 1,03. No final você ficará com 92 centavos (e não R$ 0,92).

Agora veja as recomendações do manual do PHP

O tamanho de um float depende também da plataforma e é de 64bits no formato IEEE(*). Nunca compare números em ponto flutuante em igualdades, sob pena de cometer erros.

Teste com PostgreSQL

SELECT CAST((0.1 + 0.7)*10 AS INTEGER);

Este sim, retorna o valor esperado.

Em Java:

System.out.println(1.95 - 1.03); // Retorna errado e em PHP retorna OK.

Em Ruby

(1.8+0.1)==(1.9) retorna false

O mesmo ocorre em Phyton.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/N%C3%BAmeros"

Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Strings

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 substr -- Retorna uma parte de uma string
- 2 substr_replace
- 3 Encontrar Posição de caractere em String
- 4 Contando Ocorrências de Substring em String
- 5 Trocando Ponto por Vírgula e vice-versa
- 6 Conversão de Strings
- 7 Trabalhando com os Caracteres de Strings
- 8 Validação de Caracteres
- 9 ctype_alnum - Checa por caracteres alfanuméricos
- 10 ctype_alpha - Checa por caracteres alfabéticos
- 11 ctype_digit - Checa por caracteres numéricos
- 12 ctype_lower - Checa por caracteres minúsculos
- 13 ctype_punct - Checa por Caracteres que não sejam espaço em branco nem alfanuméricos
- 14 ctype_space - Checa por espaços em branco
- 15 Validação de Tipos
- 16 Cases
- 17 Índices com Str_Pad
- 18 String para TimeStamp

substr -- Retorna uma parte de uma string

string substr (string string, int start [, int length])

Exemplo 1. Uso básico de substr()

```
<?php
$rest = substr("abcdef", 1);    // retorna "bcdef"
$rest = substr("abcdef", 1, 3); // retorna "bcd"
$rest = substr("abcdef", 0, 4); // retorna "abcd"
$rest = substr("abcdef", 0, 8); // retorna "abcdef"

// Outra opção é acessar atravéz de chaves
$string = 'abcdef';
echo $string{0};                // retorna a
echo $string{3};                // retorna d
?>
```

Se start for negativo, a string retornada irá começar no caractere start a partir do fim de string.

Exemplo 2. Usando um inicio negativo

```
<?php
$rest = substr("abcdef", -1);    // retorna "f"
$rest = substr("abcdef", -2);    // retorna "ef"
$rest = substr("abcdef", -3, 1); // retorna "d"
?>
```

Exemplo 3. Usando um length negativo

```
<?php
$rest = substr("abcdef", 0, -1);  // retorna "abcde"
$rest = substr("abcdef", 2, -1);  // retorna "cde"
$rest = substr("abcdef", 4, -4);  // retorna ""
$rest = substr("abcdef", -3, -1); // retorna "de"
?>

<h2>Sobrescrevendo Strings</h2>

str_replace

str_replace --  Substitui todas as ocorrências da string de procura com a string
de substituição

mixed str_replace ( mixed pesquisa, mixed substitui, mixed assunto [, int
&count] )

<pre>
<?php
// Fornece: <body text='black'>
$bodytag = str_replace("%body%", "black", "<body text='%body%'>");

// Fornece: Hll Wrld f PHP
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");
```

```
// Fornece: você comeria pizza, cerveja e sorvete todos os dias
$frase  = "você comeria frutas, vegetais, e fibra todos os dias.";
$saudavel = array("frutas", "vegetais", "fibra");
$saboroso  = array("pizza", "cerveja", "sorvete");

$novafrase = str_replace($saudavel, $saboroso, $frase);

// Uso do parâmetro count está disponível no PHP 5.0.0
$str = str_replace("ll", "", "good golly miss molly!", $count);
echo $count; // 2
?>
```

substr_replace

substr_replace -- Substitui o texto dentro de uma parte de uma string

string substr_replace (string string, string replacement, int start [, int length])

```
<?php
$var = 'ABCDEFGH:/MNRPQR/';
echo "Original: $var<hr>\n";

/* Estes dois exemplos substituem tudo de $var com 'bob'. */
echo substr_replace($var, 'bob', 0) . "<br>\n";
echo substr_replace($var, 'bob', 0, strlen($var)) . "<br>\n";

/* Insere 'bob' direto no começo de $var. */
echo substr_replace($var, 'bob', 0, 0) . "<br>\n";

/* Estes dois exemplos substituem 'MNRPQR' em $var com 'bob'. */
echo substr_replace($var, 'bob', 10, -1) . "<br>\n";
echo substr_replace($var, 'bob', -7, -1) . "<br>\n";

/* Deleta 'MNRPQR' de $var. */
echo substr_replace($var, '', 10, -1) . "<br>\n";
?>
```

Encontrar Posição de caractere em String

strpos

strpos -- Encontra a posição da primeira ocorrência de uma string

int strpos (string str, string procurar [, int offset])

Exemplos strpos()

```
<?php
//$str = 'abc';
$str = 'cba';
$procurar  = 'a';
$posicao = strpos($str, $procurar);

// Note o uso de ===.  Simples == não funcionaria como esperado
// por causa da posição de 'a' é 0 (primeiro) caractere.
```

```
if ($pos === false) {
   echo "A string '$procurar' não foi encontrada na string '$str'";
} else {
   echo "A string '$procurar' foi encontrada na string '$str'";
   echo " e está na posição $posicao";
}

?>

<?php

//$email = 'ribafs@gmail.com.br';
$email = 'ribafs@gmail.com';
$usuario = substr ($email, 0, strpos ($email, '@'));
// Lembrando: substr ( string string, int start [, int length] )
$dominio = substr ($email, strpos ($email, '@')+1);
echo "Usuário '$usuario' e Domínio '$dominio'"; // o comprimento default é até o
final
?>
```

Contando Ocorrências de Substring em String

substr_count -- Conta o número de ocorrências de uma substring

int substr_count (string str, string conte_me)

substr_count() retorna o número de vezes que a substring conte_me ocorre na string str.

```
<?php
  $str = "Olá mundo do PHP";

  if (substr_count($str, "do") == 0)
   echo "nenhum";

  // same as:

  if (strpos($str, "do") === false)
   echo "nenhum";
?>
```

Exemplo 1. Exemplo substr_count()

```
<?php
print substr_count("This is a test", "is"); // mostra 2
?>
```

Trocando Ponto por Vírgula e vice-versa

Se temos campos tipo moeda, devemos exibir com vírgula e gravar no banco com ponto.

Para isso uma boa saída é usar a dupla de funções implode e explode.

Antes de exibir na tela (em consultas):

$f_custo_produtivo=explode(".",$f_custo_produtivo);

$f_custo_produtivo=implode(",",$f_custo_produtivo);

Antes de gravar no banco (inclusão e atualização):

$f_custo_produtivo=explode(",",$f_custo_produtivo);

$f_custo_produtivo=implode(".",$f_custo_produtivo);

Conversão de Strings

```
$foo = 1 + "10.5";echo $foo."<br>";             // $foo é float (11.5)
$foo = 1 + "-1.3e3";echo $foo."<br>";              // $foo é float (-1299)
$foo = 1 + "bob-1.3e3";echo $foo."<br>";          // $foo é integer (1)
$foo = 1 + "bob3";echo $foo."<br>";                // $foo é integer (1)
$foo = 1 + "10 Small Pigs";echo $foo."<br>";      // $foo é integer (11)
$foo = 4 + "10.2 Little Piggies";echo $foo."<br>"; // $foo é float (14.2)
$foo = "10.0 pigs " + 1;echo $foo."<br>";          // $foo é float (11)
$foo = "10.0 pigs " + 1.0;echo $foo."<br>";        // $foo é float (11)
```

Trabalhando com os Caracteres de Strings

```
// Pega o primeiro caracter da string
$str = 'Isto é um teste.';
$first = $str{0};
echo $first."<br>";
// Pega o terceiro caracter da string
$third = $str{2};
echo $third."<br>";
// Pega o último caracter da string
$str = 'Isto ainda é um teste.';
$last = $str{strlen($str)-1};
echo $last."<br>";
// Modifica o ultimo caracter da string
$str = 'Olhe o mal';
echo $str{strlen($str)-1} = 'r';
```

Validação de Caracteres

```
ctype_alnum
ctype_alpha
ctype_cntrl
```

```
ctype_digit
ctype_graph
ctype_lower
ctype_print
ctype_punct
ctype_space
ctype_upper
ctype_xdigit
```

ctype_alnum - Checa por caracteres alfanuméricos

```
$strings = array('AbCd1zyZ9', 'foo!#$bar');

foreach ($strings as $testcase) {
   if (ctype_alnum($testcase)) {
       echo "The string $testcase consists of all letters or digits.\n";
   } else {
       echo "The string $testcase does not consist of all letters or digits.\n";
   }
}
```

ctype_alpha - Checa por caracteres alfabéticos

```
$strings = array('KjgWZC', 'arf12');
foreach ($strings as $testcase) {
   if (ctype_alpha($testcase)) {
       echo "The string $testcase consists of all letters.\n";
   } else {
       echo "The string $testcase does not consist of all letters.\n";
   }
}
```

ctype_digit - Checa por caracteres numéricos

```
$strings = array('1820.20', '10002', 'wsl!12');
foreach ($strings as $testcase) {
   if (ctype_digit($testcase)) {
       echo "The string $testcase consists of all digits.\n";
   } else {
       echo "The string $testcase does not consist of all digits.\n";
   }
}
// Alerta: Ao executar veja que somente é válido quando todos são dígitos
// Não é indicado para testar valores decimais, com ponto ou vírgula
```

ctype_lower - Checa por caracteres minúsculos

```
$strings = array('aac123', 'qiutoas', 'QASsdks');
foreach ($strings as $testcase) {
   if (ctype_lower($testcase)) {
       echo "The string $testcase consists of all lowercase letters.\n";
   } else {
```

```
        echo "The string $testcase does not consist of all lowercase letters.\n";
    }
}
```

ctype_punct - Checa por Caracteres que não sejam espaço em branco nem alfanuméricos

```
$strings = array('ABasdk!@!$#', '!@ # $', '*&$()');
foreach ($strings as $testcase) {
    if (ctype_punct($testcase)) {
        echo "The string $testcase consists of all punctuation.\n";
    } else {
        echo "The string $testcase does not consist of all punctuation.\n";
    }
}
```

ctype_space - Checa por espaços em branco

Validação de Tipos

```
intval
is_array
is_bool
is_callable
is_double
is_float
is_int
is_integer
is_long
is_null
is_numeric
is_object
is_real
is_resource
is_scalar
is_string
isset
print_r
serialize
settype
strval
unserialize
unset
```

Cases

```
strtoupper($str) - tudo maiúsculo
strtolower($str) - tudo minúsculo
ucfirst($str) - Converte para maiúscula o primeiro caractere de uma STRING
ucwords($STR) - Converte para maiúsculas o primeiro caractere de cada PALAVRA
```

Índices com Str_Pad

str_pad -- Preenche uma string para um certo tamanho com outra string

string str_pad (string input, int pad_length [, string pad_string [, int pad_type]])

Exemplo:

$players =

```
  array("DUNCAN, king of Scotland"=>"Larry",
        "MALCOLM, son of the king"=>"Curly",
        "MACBETH"=>"Moe",
        "MACDUFF"=>"Rafael");

echo "
";

// Print a heading
echo str_pad("Dramatis Personae", 50, " ", STR_PAD_BOTH) . "\n";

// Print an index line for each entry
foreach($players as $role=>$actor)
  echo str_pad($role, 30, ".")
      . str_pad($actor, 20, ".", STR_PAD_LEFT) . "\n";

echo "

";
```

Resultado:

```
                    Dramatis Personae
DUNCAN, king of Scotland.....................Larry
MALCOLM, son of the king.....................Curly
MACBETH.......................................Moe
MACDUFF.....................................Rafael
```

String para TimeStamp

```
// Absolute dates and times
$var = strtotime("25 December 2002");
$var = strtotime("14/5/1955");
$var = strtotime("Fr1, 7 Sep 2001 10:28:07 -1000");
```

```
// The current time: equivalent to time(  )
$var = strtotime("now");

// Relative times
echo strtotime("+1 day");
echo strtotime("-2 weeks");
echo strtotime("+2 hours 2 seconds");

//Care should be taken when using strtotime( ) with user-supplied dates. It's
better to limit the use of strtotime( ) to cases when
//the string to be parsed is under the control of the script, for example,
checking a minimum age using a relative date:
// date of birth: timestamp for 16 August, 1983
$dob = mktime(0, 0, 0, 16, 8, 1982);

// Now check that the individual is over 18
if ((float)$dob < (float)strtotime("-18 years"))
  echo "Legal to drive in the state of Victoria";
```

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Strings"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Permissões de arquivos e diretórios

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 chmod - altera permissões de arquivos e diretórios
- 2 chown
- 3 chgrp -- Modifica o grupo do arquivo
- 4 is_writable -- Diz se pode-se escrever para o arquivo (writable)
- 5 umask -- Modificar a umask atual

chmod - altera permissões de arquivos e diretórios

```
<?php
chmod ("/arquivo/diretorio", 755);  // decimal; provavelmente incorreto
chmod ("/arquivo/diretorio", "u+rwx,go+rx"); // string; incorreto
chmod ("/arquivo/diretorio", 0755);  // octal; representaÃ§Ã£o correta do modo
?>

function permissoes($arquivo,$perms,$acao){
        print "<form name=frm method=post action=acoes.php>";
        print "<input name=pm value=$perms>";
        print "<input type=hidden name=perms value=$perms>";
        print "<input type=hidden name=ar value=$arquivo>";
        print "<input type=hidden name=acao value=$acao>";
        print "<input name=ar value=$arquivo readonly style='background-
color:#FAEBD7'>";
        print "<input type=submit name=prm value=Alterar>";
        print "</form>";

        if (isset($_POST['prm'])){
                $ar=$_POST['ar'];
                $perms=octdec($_POST['pm']);
                $ch = chmod($ar, $perms);
                if(!$ch) {
                        die ("Erro ao alterar as permissÃ□Âµes!");
                }else{
                        print "<script>location='index.php'</script>";
                }
        }
}
```

```
<?php
// Escrita e leitura para o proprietario, nada ninguem mais
chmod ("/somedir/somefile", 0600);

// Escrita e leitura para o proprietario, leitura para todos os outros
chmod ("/somedir/somefile", 0644);

// Tudo para o proprietario, leitura e execucao para os outros
chmod ("/somedir/somefile", 0755);

// Tudo para o proprietario, leitura e execucao para o grupo do prop
chmod ("/somedir/somefile", 0750);
?>

Value    Permission Level
400    Owner Read
200    Owner Write
100    Owner Execute
40    Group Read
20    Group Write
10    Group Execute
4    Global Read
2    Global Write
1    Global Execute


<?php
function chmodnum($mode) {
   $mode2=$mode;
   $realmode = "";
   $legal =  array("","w","r","x","-");
   $attarray = preg_split("//",$mode);
   for($i=0;$i<count($attarray);$i++){
       if($key = array_search($attarray[$i],$legal)){
           $realmode .= $legal[$key];
       }
   }
   $mode = str_pad($realmode,9,'-');
   $trans = array('-'=>'0','r'=>'4','w'=>'2','x'=>'1');
   $mode = strtr($mode,$trans);
   $newmode = '';
   $newmode .= $mode[0]+$mode[1]+$mode[2];
   $newmode .= $mode[3]+$mode[4]+$mode[5];
   $newmode .= $mode[6]+$mode[7]+$mode[8];
   return $mode2.' = '.$newmode;
}

echo chmodnum('drwxr-xr-x');
?>

alguns exemplos:

drwxr-xr-x => 755
drwxr-xr-x => 755
dr-xr-xr-x => 555
drwxr-xr-x => 755
drwxr-xr-x => 755
drwxr-xr-x => 755
drwxr-xr-x => 755
```

```
drwxrwxrwt => 776
drwxr-xr-x => 755
drwxr-xr-x => 755
lrwxrwxrwx => 777
```

chown

Esta função não trabalha com arquivos remotos

```
<?php

$file_name= "test";
$path = "/var/www/html/test/" . $file_name ;

$user_name = "root";

chown($path, $user_name);

?>

<?php
function recurse_chown_chgrp($mypath, $uid, $gid)
{
   $d = opendir ($mypath) ;
   while(($file = readdir($d)) !== false) {
       if ($file != "." && $file != "..") {

           $typepath = $mypath . "/" . $file ;

           //print $typepath. " : " . filetype ($typepath). "<BR>" ;
           if (filetype ($typepath) == 'dir') {
               recurse_chown_chgrp ($typepath, $uid, $gid);
           }

           chown($typepath, $uid);
           chgrp($typepath, $gid);

       }
   }

 }

recurse_chown_chgrp ("uploads", "ribafs", "meugrupo") ;
?>

<?php
function recurse_chown_chgrp($path2dir, $uid, $gid){
   $dir = new dir($path2dir);
   while(($file = $dir->read()) !== false) {
       if(is_dir($dir->path.$file)) {
           recurse_chown_chgrp($dir->path.$file, $uid, $gid);
       } else {
           chown($file, $uid);
           chgrp($file, $gid);
       }
   }
```

```
   $dir->close();
}
?>
```

chgrp -- Modifica o grupo do arquivo

filegroup -- Lê o grupo do arquivo

fileperms -- Lê as permissões do arquivo

fileowner -- Lê o dono (owner) do arquivo

is_readable -- Diz se o arquivo/diretório é legivel (readable)

```
<?php
if (is_readable('my_link')) {
  header('Location: /my_link');
}
?>
```

is_writable -- Diz se pode-se escrever para o arquivo (writable)

```
<?php

$file = '/home/vincent/arquivo.sh';

if(is_executable($file)) {
   echo $file.' Ã© executÃ¡vel';
} else {
   echo $file.' nÃ£o Ã© executÃ¡vel';
}

?>
```

umask -- Modificar a umask atual

```
<?php
umask(0670);                     //- set umask
$handle = fopen('file', 'w');  //- 0006
mkdir("/path/dir");            //- 0107
?>

calculate the result:
<?php
$umask = 0670;
umask($umask);
//- if you are creating a new directory, $permission = 0777;
//- if you are creating a new file, $permission = 0666.
printf( "result: %04o", $permission & ( 0777 - $umask) );
?>
```

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Permiss%C3%B5es_de_arquivos_e_diret%C3%B3rios"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Validação de Dados

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 Ótimo artigo em 5 partes na Revista do PHP de autoria do Er Abbott
- 2 Validação de e-mails
  - 2.1 Check valid e-mail
- 3 Tipos de Variáveis

Ótimo artigo em 5 partes na Revista do PHP de autoria do Er Abbott

- Validação de Formulários - 5º e última parte: Validando no servidor

- Validação de Formulários - Parte 04 (A fonteira cliente/servidor)

- Validação de Formulários - Parte 3: O baile de máscaras

- Validação de Formulários - Parte 2: Os Campos Especiais

- Validação de formulários - Parte 1 (O Planejamento)

Turbinando a Validação de Formulários

Validação de e-mails

Check valid e-mail

```
function esEmailValido($email)
{
    if (ereg("^[_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)*@([_a-zA-Z0-9-]+\.)*[a-zA-Z0-9-]
{2,200}\.[a-zA-Z]{2,6}$", $email ) )
        {
       return true;
    }
        else
        {
       return false;
```

```
    }
}
```

http://snippets.dzone.com/posts/show/4346

Tipos de Variáveis

```
<?php

function ss_array_as_string (&$array, $column = 0) {
    $str = "Array(<BR>\n";
    while(list($var, $val) = each($array)){
        for ($i = 0; $i < $column+1; $i++){
            $str .= "    ";
        }
        $str .= $var.' ==> ';
        $str .= ss_as_string($val, $column+1)."<BR>\n";
    }
    for ($i = 0; $i < $column; $i++){
        $str .= "    ";
    }
    return $str.')';
}

function ss_object_as_string (&$object, $column = 0) {
    if (empty($object->classname)) {
        return "$object";
    }
    else {
        $str = $object->classname."(<BR>\n";
        while (list(,$var) = each($object->persistent_slots)) {
            for ($i = 0; $i < $column; $i++){
                $str .= "    ";
            }
            global $$var;
            $str .= $var.' ==> ';
            $str .= ss_as_string($$var, column+1)."<BR>\n";
        }
        for ($i = 0; $i < $column; $i++){
            $str .= "    ";
        }
        return $str.')';
    }
}

function ss_as_string (&$thing, $column = 0) {
    if (is_object($thing)) {
        return ss_object_as_string($thing, $column);
    }
    elseif (is_array($thing)) {
        return ss_array_as_string($thing, $column);
    }
    elseif (is_double($thing)) {
        return "Double(".$thing.")";
    }
    elseif (is_long($thing)) {
```

```
        return "Long(".$thing.")";
    }
    elseif (is_string($thing)) {
        return "String(".$thing.")";
    }
    else {
        return "Unknown(".$thing.")";
    }
}

$my_variable=3;
//echo ss_as_string($my_variable);
echo ss_as_string($GLOBALS);
?>
```

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Valida%C3%A7%C3%A3o_de_Dados"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Arrays

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 Trabalhando com Arrays
- 2 Algumas das funções
- 3 array_fill -- Preenche um array com valores
- 4 array_merge -- Funde dois ou mais arrays
- 5 array_pad -- Expande um array para um certo comprimento utilizando um determinado valor
- 6 array_pop -- Retira um elemento do final do array
- 7 array_push -- Adiciona um ou mais elementos no final de um array
- 8 array_reverse -- Retorna um array com os elementos na ordem inversa
- 9 array_search
- 10 array_shift -- Retira o primeiro elemento de um array
- 11 array_sum -- Calcula a soma dos elementos de um array
- 12 array -- Cria um array
- 13 arsort
- 14 asort
- 15 count -- Conta o número de elementos de uma variável
- 16 current -- Retorna o elemento corrente em um array
- 17 each -- Retorna o par chave/valor corrente de um array e avança o seu cursor
- 18 Percorrendo um array com each()
- 19 end -- Faz o ponteiro interno de um array apontar para o seu último elemento
- 20 key -- Retorna uma chave da posição atual de um array associativo
- 21 next -- Avança o ponteiro interno de um array
- 22 prev -- Retrocede o ponteiro interno de um array
- 23 sizeof -- Apelido de count()
- 24 sort -- Ordena um array pelo seu valor
- 25 Exemplo de array multidimensional
- 26 Exemplo de Array
- 27 Convertendo objetos para um array

Trabalhando com Arrays

Um array é uma variável, mas diferente das demais ele armazena uma coleção de valores e não somente um. E ainda por cima podem conter outras variáveis e de tipos diferentes.

Detalhe importante: Quando em uma função precisarmos rtornar mais de um valor array é a saída, basta retornar todos os valors em forma de array.

Além disso é semelhante ao que estudamos na matemática: linhas e colunas. Matriz 3x4 (3 linhas e 4 colunas).

Um array no PHP é um mapa ordenado, que relaciona valores com chaves (em linhas e colunas).

Especificando um array()

array([chave =>] valor, ...);

A chave pode ser uma string ou um inteiro.

O valor pode ser qualquer coisa.

Algumas das funções

Essas funções abaixo permitem a interação e manipulação de arrays de várias formas. Arrays são essenciais para armazenar, gerenciar, operar sobre um conjunto de variáveis.

Arrays (matrizes) simples e multidimensionais são suportados e podem ser criados pelo usuário ou por outras funções. Existem diversas funções específicas para bancos de dados, que preenchem arrays com os dados retornados em consultas, e vários outros tipos de funções também retornam arrays.

array_fill -- Preenche um array com valores

array array_fill (int start_index, int num, mixed value)

```
<?php
$a = array_fill(5, 6, 'banana');
print_r($a);
?>
```

array_merge -- Funde dois ou mais arrays

array array_merge (array array1, array array2 [, array ...])

```
<?php
$array1 = array();
$array2 = array(1 => "data");
$result = array_merge($array1, $array2);
```

```
?>
```

Não esqueça que as chaves numéricas serão reordenadas!

```
Array
(
   [0] => data
)
```

Se você quer preservar os arrays e apenas concatená-los, o operador +:

```
<?php
$array1 = array();
$array2 = array(1 => "data");
$result = $array1 + $array2;
?>
```

As chaves numéricas serão preservadas e as associações originais permanecem.

array_pad -- Expande um array para um certo comprimento utilizando um determinado valor

array array_pad (array input, int pad_size, mixed pad_value)

Exemplo 1. Exemplo de array_pad()

```
<?php
$input = array(12, 10, 9);

$result = array_pad($input, 5, 0);
// $result é array(12, 10, 9, 0, 0)

$result = array_pad($input, -7, -1);
// $result é array(-1, -1, -1, -1, 12, 10, 9)

$result = array_pad($input, 2, "noop");
// Não será expandido
?>
```

array_pop -- Retira um elemento do final do array

mixed array_pop (array array)

```
<?php
$cesta = array("laranja", "banana", "melancia", "morango");
$fruta = array_pop($cesta);
print_r($cesta);
?>
```

array_push -- Adiciona um ou mais elementos no final de um array

int array_push (array array, mixed var [, mixed ...])

```
<?php
$cesta = array("laranja", "morango");
array_push($cesta, "melancia", "batata");
print_r($cesta);
?>
```

array_reverse -- Retorna um array com os elementos na ordem inversa

array array_reverse (array array [, bool preserve_keys])

```
<?php
$input = array("php", 4.0, array ("verde", "vermelho"));
$result = array_reverse($input);
$result_keyed = array_reverse($input, TRUE);
print_r($result_keyed);
?>
```

array_search

-- Procura por um valor em um array e retorna sua chave correspondente caso seja encontrado. Caso contrário retorna FALSE.

mixed array_search (mixed procurar_este, array procurar_neste [, bool strict])

```
<?php
$a=array("a","b",0,"c","d");
echo "a: ".array_search("a",$a)."<br>";
echo "b: ".array_search("b",$a)."<br>";
echo "c: ".array_search("c",$a)."<br>";
echo "d: ".array_search("d",$a)."<br>";
echo "0: ".array_search("0",$a)."<br>";
echo "x: ".array_search("x",$a)."<br>";
echo "1: ".array_search("1",$a);
?>

<?php
if (array_search($needle, $array)!== FALSE) {
   //code goes here (
}
?>

<?php

function array_replace($search, $replace, &$array) {
  foreach($array as $key => $value) {
```

```
   if($value == $search) {
     $array[$key] = $replace;
   }
  }
}
?>

<?
$Projects[0] = array(123, "Text 1");
$Projects[1] = array(456, "Text 2");
$Projects[2] = array(789, "Text 3");

$search_value = "ext 3";

foreach ($Projects as $key => $row){
     foreach($row as $cell){
       if (strpos($cell, $search_value) !== FALSE){
                echo  "<p>Project ".$key;
       }
     }
 }
?>
```

array_shift -- Retira o primeiro elemento de um array

mixed array_shift (array array)

```
<?php
$cesta = array("laranja", "banana", "melancia", "morango");
$fruta = array_shift($cesta);
print_r($cesta);
?>
```

array_sum -- Calcula a soma dos elementos de um array

mixed array_sum (array arr)

```
<?php
$a = array(2, 4, 6, 8);
echo "soma(a) = ".array_sum($a)."<br>";

$b = array("a" => 1.2, "b" => 2.3, "c" => 3.4);
echo "soma(b) = ".array_sum($b)."<br>";
?>
```

array_unique -- Remove os valores duplicados de um array

array array_unique (array array)

```
<?php
$input = array("a" => "verde", "vermelho", "b" => "verde", "azul", "vermelho");
$result = array_unique($input);
```

```
print_r($result);
?>
```

Exemplo 2. array_unique() e tipos

```
<?php
$input = array(4, "4", "3", 4, 3, "3");
$result = array_unique($input);
var_dump($result);
?>
<pre>


<h2>array_values -- Retorna todos os valores de um array</h2>

array array_values ( array input )

//Retorna os valores, as chaves não
<pre>
<?php
$array = array("tamanho" => "G", "cor" => "dourado");
print_r(array_values ($array));
?>
```

array -- Cria um array

array array ([mixed ...])

Exemplo 1. Exemplo de array()

```
<?php
$frutas = array (
   "frutas" => array("a"=>"laranja", "b"=>"banana", "c"=>"maçã"),
   "numeros" => array(1, 2, 3, 4, 5, 6),
   "buracos" => array("primeiro", 5 => "segundo", "terceiro")
)
?>
```

Exemplo 2. Indexação automática com array()

```
<?php
$array = array(1, 1, 1, 1,  1, 8 => 1,  4 => 1, 19, 3 => 13);
print_r($array);
?>
```

arsort

-- Ordena um array em ordem descrescente dos valores mantendo a associação entre índices e valores

void arsort (array array [, int sort_flags])

```
<?php
```

```
$frutas = array("d" => "limao", "a" => "laranja", "b" => "banana", "c" =>
"melancia");
arsort($frutas);
reset($frutas);
while (list($chave, $valor) = each($frutas)) {
   echo "$chave = $valor\n";
}
?>
```

asort

-- Ordena um array em ordem crescente dos valores mantendo a associação entre índices e valores

void asort (array array [, int sort_flags])

```
<?php
$frutas = array("d" => "limao", "a" => "laranja", "b" => "banana", "c" =>
"melancia");
asort($frutas);
reset($frutas);
while (list($chave, $valor) = each($frutas)) {
   echo "$chave = $valor\n";
}
?>
```

count -- Conta o número de elementos de uma variável

int count (mixed var [, int mode])

```
<?php
$a[0] = 1;
$a[1] = 3;
$a[2] = 5;
$a[3] = 6;
$result = count($a);
// $result == 4
print $result."<br>";

$b[0] = 7;
$b[5] = 9;
$b[10] = 11;
$result = count($b);
// $result == 3;
print $result;
?>
```

Exemplo 2. Uso recursivo da função count() (PHP >= 4.2.0)

```
<?php
$food = array( 'fruits'  => array('orange', 'banana', 'apple'),
'veggie'  => array('carrot', 'collard','pea'));
// recursive count
echo count($food,COUNT_RECURSIVE);  // mostra 8
```

```
// normal count
echo count($food);                  // mostra2 2
?>

<?php
$food = array( 'fruits'  => array('orange', 'banana', 'apple'),
'veggie'  => array('carrot', 'collard','pea'));
// recursive count
echo count($food,COUNT_RECURSIVE)."<br>";  // mostra 8
// normal count
echo count($food);                  // mostra2 2
?>
```

current -- Retorna o elemento corrente em um array

mixed current (array array)

```
<?php
$transport = array('foot', 'bike', 'car', 'plane');
$mode = current($transport); // $mode = 'foot';
echo "Atual $mode<br>";
$mode = next($transport);    // $mode = 'bike';
echo "Atual $mode<br>";
$mode = current($transport); // $mode = 'bike';
echo "Atual $mode<br>";
$mode = prev($transport);    // $mode = 'foot';
echo "Atual $mode<br>";
$mode = end($transport);    // $mode = 'plane';
echo "Atual $mode<br>";
$mode = current($transport); // $mode = 'plane';
echo "Atual $mode<br>";
?>
```

each -- Retorna o par chave/valor corrente de um array e avança o seu cursor

array each (array array)

```
<?php
$foo = array("bob", "fred", "jussi", "jouni", "egon", "marliese");
$bar = each($foo);
print_r($bar);
?>

<?php
  $foo = array("Robert" => "Bob", "Seppo" => "Sepi");
  $bar = each($foo);
  print_r($bar);
?>
```

Percorrendo um array com each()

```
<?php
$fruit = array('a' => 'apple', 'b' => 'banana', 'c' => 'cranberry');
reset($fruit);
while (list($key, $val) = each($fruit)) {
     echo "$key => $val\n";
}
/* Saída:

  a => apple
  b => banana
  c => cranberry

  */
?>
```

end -- Faz o ponteiro interno de um array apontar para o seu último elemento

mixed end (array array)

```
<?php
  $frutas = array('melancia', 'banana', 'morango');
  print end($frutas); // morango
?>
```

key -- Retorna uma chave da posição atual de um array associativo

mixed key (array array)

```
<?php
  $array = array(
     'fruit1' => 'apple',
     'fruit2' => 'orange',
     'fruit3' => 'grape',
     'fruit4' => 'apple',
     'fruit5' => 'apple');

  // este ciclo  exibirá todas as chaves do array associativo
  // ausiliado pela função next()
  while ($fruit_name = current($array)) {
         echo key($array).'<br>';
             next($array);
  }
  ?>

<?php
  $array = array(
```

```
    'fruit1' => 'apple',
    'fruit2' => 'orange',
    'fruit3' => 'grape',
    'fruit4' => 'apple',
    'fruit5' => 'apple');

  // este ciclo  exibirá toda a chave do array associativo
  // onde o valor é igual a "apple"
  while ($fruit_name = current($array)) {
     if ($fruit_name == 'apple') {
         echo key($array).'<br>';
     }
     next($array);
  }
  ?>
```

next -- Avança o ponteiro interno de um array

mixed next (array array)

```
<?php
  $transport = array('foot', 'bike', 'car', 'plane');
  $mode = current($transport); // $mode = 'foot';
  print"$mode<br>";
  $mode = next($transport);    // $mode = 'bike';
  print"$mode<br>";
  $mode = next($transport);    // $mode = 'car';
  print"$mode<br>";
  $mode = prev($transport);    // $mode = 'bike';
  print"$mode<br>";
  $mode = end($transport);    // $mode = 'plane';
  print"$mode<br>";
?>
```

prev -- Retrocede o ponteiro interno de um array

mixed prev (array array)

```
<?php
  $transport = array('foot', 'bike', 'car', 'plane');
  $mode = current($transport); // $mode = 'foot';
print"$mode<br>";
  $mode = next($transport);    // $mode = 'bike';
print"$mode<br>";
  $mode = next($transport);    // $mode = 'car';
print"$mode<br>";
  $mode = prev($transport);    // $mode = 'bike';
print"$mode<br>";
  $mode = end($transport);    // $mode = 'plane';
print"$mode<br>";
?>
<pre>
```

```
<h2>reset --  Faz o ponteiro interno de um array apontar para o seu primeiro
elemento</h2>

mixed reset ( array array )
<pre>
<?php
  $array = array('primero passo', 'segundo passo', 'terceiro passo', 'quarto
passo');

  // por definição, o ponteiro está sobre o primeiro elemento
  echo current($array)."<br>\n"; // "Primeiro passo"

  // pula dois passos
  next($array);
  next($array);
  echo current($array)."<br>\n"; // "passo três"

  // reinicia o ponteiro, começa novamente o primeiro passo
  reset($array);
  echo "Depois de resetado...: " . current($array)."<br>\n"; // "primeiro passo"
?>
```

sizeof -- Apelido de count()

sort -- Ordena um array pelo seu valor

void sort (array array [, int sort_flags])

```
<?php

$frutas = array("limao", "laranja", "banana", "melancia");
sort($frutas);
reset($frutas);
while (list($chave, $valor) = each($frutas)) {
   echo "frutas[".$chave."] = ".$valor."<br>";
}
?>
```

Os seguintes também são funcionalmente idênticos:

```
<?php
$arr = array("one", "two", "three");
reset($arr);
while (list($key, $value) = each ($arr)) {
   echo "Chave: $key; Valor: $value<br />\n";
}

foreach ($arr as $key => $value) {
   echo "Chave: $key; Valor: $value<br />\n";
}
```

```
?>
```

Mais alguns exemplos para demonstrar os usos:

```
<?php
/* exemplo foreach 1: somente valores */

$a = array(1, 2, 3, 17);

foreach ($a as $v) {
   echo "Valor atual de \$a: $v.\n";
}

/* exemplo foreach 2: valores (com as chaves impressas para ilustração) */

$a = array(1, 2, 3, 17);

$i = 0; /* para exemplo somente */

foreach ($a as $v) {
   echo "\$a[$i] => $v.\n";
   $i++;
}

/* exemplo foreach 3: chaves e valores */

$a = array (
   "um" => 1,
   "dois" => 2,
   "três" => 3,
   "dezessete" => 17
);

foreach ($a as $k => $v) {
   echo "\$a[$k] => $v.\n";
}

/* exemplo foreach 4: arrays multidimensionais */

$a[0][0] = "a";
$a[0][1] = "b";
$a[1][0] = "y";
$a[1][1] = "z";

foreach ($a as $v1) {
   foreach ($v1 as $v2) {
       echo "$v2\n";
   }
}

/* exemplo foreach 5: arrays dinâmicos */

foreach (array(1, 2, 3, 4, 5) as $v) {
   echo "$v\n";
}
?>
```

Exemplo de array multidimensional

```
$produto[1][codigo] = "1";
$produto[1][nome] = "João Pereira Brito";
$produto[1][email] = "joao@joao.org";
$produto[1][rua] = "Vasco da Gama";
$produto[1][numero] = "1345";

$produto[2][codigo] = "2";
$produto[2][nome] = "Antônio queiroz";
```

Exemplo de Array

```
$i=0;
while($i < $numregs){
        $codigo=pg_result($consulta,$i,codigo);
        $nome=pg_result($consulta,$i,nome);
        $venc=pg_result($consulta,$i,vencimento);
        $apartamento=pg_result($consulta,$i,apartamento);
        $pessoas=pg_result($consulta,$i,pessoas);
        $cota_agua=pg_result($consulta,$i,cota_agua);
        $cota_condominio=pg_result($consulta,$i,cota_condominio);
        $cota_reserva=pg_result($consulta,$i,cota_reserva);

        $total = $cota_agua + $cota_condominio + $cota_reserva;
        $total = number_format($total,2, ',','.');
...
$i++;
}
```

Também podemos ter um array formado por outros arrays (neste caso, cada sub array é uma linha do principal)

```
$arrayvarios = array(
        array(1, 3, 5, 7),
        array(2, 4, 6, 8),
        array(1, 1, 1, 1)
);
```

Neste caso temos um array 2x4 (2 linhas por 4 colunas, que iniciam sempre com índice zero).

Então se queremos retornar o valor 8, que está na linha 2 e coluna 4, devemos retornar o índice 1,3 (linha2=índice 1, coluna4=índice3).

print $arrayvarios[1][3];

Agora veremos com detalhes os pares: chave => valor:

```
$alunos = array(
        "0732355" => "Ribamar FS",
        "0823456" => "Antônio Brito",
        "0654345" => "Roberto Queiroz"
);
```

O que isto retornaria?

```
print $alunos["0732355"];
```

```
print $alunos[0];
```

Experimente!!

Atribuindo valores às chaves de arrays

Também podemos fazer diretamente assim:

print $alunos["0732355"] = "João Brito";

Lembrando que, a chave, é exclusiva. Podemos ter

```
$alunos["0732355"] = "João Brito";
$alunos["0932355"] = "João Brito";

Mas não podemos ter:
$alunos["0732355"] = "João Brito";
$alunos["0732355"] = "Ribamar FS";
```

Anexo agora um excelente tutorial sobre Arrays do Celso Goya publicado em:
http://www.xoopstotal.com.br/modules/wfsection/article.php?articleid=51

Trabalhando com arrays

Visão geral Para facilitar o entendimento, vamos definir array como um conjunto de valores, que podem ser identificados em grupo ou então separadamente. Estes conjuntos podem ser muito úteis enquanto programamos, pois em alguns casos podem substituir uma tabela em banco de dados ou então utilizando métodos mais avançados podemos carregá-los dinâmicamente e utilizar quase como um banco de dados em memória.

A linguagem PHP oferece uma incrível gama de recursos para se trabalhar com arrays. Com destaque para as funções auxiliares que permitem fazer desde uma simples contagem de elementos até a conversão automática de um array em string.

Neste artigo desenvolveremos como exemplo uma função para gerar combo boxes com os estados do Brasil. Muitas vezes criamos uma tabela no banco de dados para armazenar a lista de estados do Brasil sendo que neste caso existe um número finito de registros e menor que 100, então as operações de banco de dados não são tão ágeis quanto o uso de um array.

Criando o primeiro array

Para utilizar um array, antes de mais nada é preciso criar uma variável do tipo array.

```
<?php
$estados = array();
?>
```

O próximo passo é montar nossa lista de estados.

```
<?php
$estados = array();
$estados[0] = "Acre";
$estados[1] = "Alagoas";
$estados[2] = "Amapá";
$estados[3] = "Amazonas";
?>
```

Os colchetes servem para identificar qual elemento do nosso conjunto estamos nos referindo e o número entre colchetes é o código identificador do elemento.

Podemos fazer o seguinte teste:

```
<?php
$estados = array();
$estados[0] = "Acre";
$estados[1] = "Alagoas";
$estados[2] = "Amapá";
$estados[3] = "Amazonas";
echo($estados[0]);
?>
```

Neste caso será exibida a palavra Acre, pois indicamos o ítem [0] da variável $estados, que é um array.

Você deve estar se perguntando "O que há de tão fantástico em um array?". Agora vamos mostrar alguns recursos.

Criando o array de estados

Nosso array não será de grande valia se não permitir que as siglas dos estados sejam armazenadas também, pois desta forma podemos guardar no banco de dados apenas os dois caractéres correspondentes à sigla do estado, ou seja, utilizaremos apenas dois bytes no banco de dados.

Então vamos criar um array com duas colunas, sendo a primeira a sigla do estado e a segunda seu nome por extenso.

```
<?php
$estados = array();
$estado[0][0] = "AC";
$estado[0][1] = "Acre";
$estado[1][0] = "AL";
$estado[1][1] = "Alagoas";
$estado[2][0] = "AP";
$estado[2][1] = "Amapá";
$estado[3][0] = "AM";
$estado[3][1] = "Amazonas";
$estado[4][0] = "BA";
$estado[4][1] = "Bahia";
$estado[5][0] = "CE";
$estado[5][1] = "Ceará";
$estado[6][0] = "DF";
$estado[6][1] = "Distrito Federal";
$estado[7][0] = "ES";
$estado[7][1] = "Espírito Santo";
$estado[8][0] = "GO";
$estado[8][1] = "Goiás";
$estado[9][0] = "MA";
$estado[9][1] = "Maranhão";
$estado[10][0] = "MG";
$estado[10][1] = "Minas Gerais";
$estado[11][0] = "MT";
$estado[11][1] = "Mato Grosso";
$estado[12][0] = "MS";
$estado[12][1] = "Mato Grosso do Sul";
$estado[13][0] = "PA";
```

```
$estado[13][1] = "Pará";
$estado[14][0] = "PR";
$estado[14][1] = "Paraná";
$estado[15][0] = "PE";
$estado[15][1] = "Pernambuco";
$estado[16][0] = "PI";
$estado[16][1] = "Piauí";
$estado[17][0] = "RJ";
$estado[17][1] = "Rio de Janeiro";
$estado[18][0] = "RN";
$estado[18][1] = "Rio Grande do Norte";
$estado[19][0] = "RS";
$estado[19][1] = "Rio Grande do Sul";
$estado[20][0] = "RO";
$estado[20][1] = "Rondônia";
$estado[21][0] = "RR";
$estado[21][1] = "Roraima";
$estado[22][0] = "SC";
$estado[22][1] = "Santa Catarina";
$estado[23][0] = "SP";
$estado[23][1] = "São Paulo";
$estado[24][0] = "SE";
$estado[24][1] = "Sergipe";
$estado[25][0] = "TO";
$estado[25][1] = "Tocantins";
?>
```

A diferença neste exemplo é que utilizamos dois identificadores de elemento, ou seja, agora para cada elemento do array possuimos mais outros dois dependentes. Da mesma forma que criamos dois elementos o 0 e 1 para cada item de estado poderiamos criar n novos sub-elementos, por exemplo:

```
<?php
$estado = array();

$estado[0][0] = "SP";
$estado[0][1] = "São Paulo";
$estado[0][2] = "Sudeste";
?>
```

Vamos considerar à partir de agora que um array possui linhas e colunas, onde as linhas são equivalentes ao primeiro conjunto de colchetes e as colunas são equivalentes ao segundo conjunto de colchetes.

A função de exibição do combo box de estados

Agora vamos exibir todos os elementos de nosso array em uma função:

```
<?php
/*Nossa função reebe 3 parâmetros
$pNome :: Corresponde ao nome do SELECT
$pSelected :: Corresponde ao elemento que deverá possuir o status de selecionado
automaticamente
$extra :: Caso precise adicionar um style, ou então opção de multiple
*/
function renderCombo($pNome = "", $pSelected = "SP", $extra = ""){
        echo("<SELECT NAME='".$pNome."' ".$extra.">");
```

```
/*
Para exibir todos os itens do nosso combo utilizamos o comando for ,
lembre-se que como usamos números para identificar nosso array,
então podemos substituí-lo automaticamente com o for
*/
        //Realiza o loop em todos os elementos do array
        for( $i = 0; $i < 26;$i++ ){
                //Imprime a TAG OPTION usando a primeira coluna do array
                echo("<OPTION VALUE='".$estado[$i][0]."'");
                //Efetua a comparação para verificar se este é o ítem
selecionado
                if( $estado[$i][0] == $pSelected ){
                        //Caso a comparação seja verdadeira seleciona o ítem
                        echo(" SELECTED");
                }
                //Imprime o nome por extenso do estado, equilavente a segunda
coluna do array
                echo(">".$estado[$i][1]."</option>\n");
        }
        //Finaliza a tag SELECT
        echo("</SELECT>\n");
}
?>
```

Eureka! Esta feita uma função para exibir um combo de estados.

Identificadores alternativos

Na linguagem PHP podemos utilizar palavras para identificar um elemento de um array, este recurso é muito bom, pois facilita muito a depuração e o entendimento de programas que utilizam arrays.

Vamos utilizar nosso array de estados como exemplo:

```
<?php
$estado[0]["sigla"] = "SP";
$estado[0]["nome"] = "São Paulo";
$estado[0]["regiao"] = "Sudeste";

echo($estado[0]["sigla"]);
?>
```

Desta forma podemos deixar o código de nossos programas mais fáceis de se compreender. Repare que utilizamos uma string simples para identificar um elemento do array, sendo assim, podemos utilizar variáveis para identificar um item do array, por exemplo:

```
<?php
$estado[0]["sigla"] = "SP";
$estado[0]["nome"] = "São Paulo";
$estado[0]["regiao"] = "Sudeste";

$variavel = "sigla";

echo($estado[0][$variavel]);
?>
```

É importante lembrar que mesmo existindo uma string para identificar um elemento do array ainda podemos utilizar números se quisermos, por exemplo:

```
<?php
$estado[0]["sigla"] = "SP";
$estado[0]["nome"] = "São Paulo";
$estado[0]["regiao"] = "Sudeste";

echo($estado[0][0]);
?>
```

Nos três casos o resultado é o mesmo, diferindo apenas no método como chamamos o array.

O que você viu neste artigo é o básico sobre arrays, caso você se interessar pelo assunto e queira dar uma pesquisada rápida na web, vai encontrar outras formas de declarar arrays bem como usos diferenciados. O XOOPS utiliza muito este recurso. É só dar uma olhada em algum arquivo xoops_version.php, que você vai encontrar um exemplo prático do uso de arrays.

Final do tutorial do Celso Goya.

Convertendo objetos para um array

http://www.revistaphp.com.br/print.php?id=147

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Arrays"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/XML

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Trabalhando com XML em PHP

Ler Arquivo com XML

Observação: A função simplexml_load_string está apenas disponível no **PHP5**.

```
<?php
include 'xml_lido.php'; // arquivo a ser lido

$xml = simplexml_load_string($xmlstr);

echo $xml->filme[0]->comentario;
print '<br>';
echo $xml->filme[0]->titulo;
print '<br>';
echo $xml->filme[0]->personagens[0]->personagem[0]->nome;
print '<br>';
echo $xml->filme[0]->votos[0];
print '<br>';
echo $xml->filme[0]->votos[1];
?>
```

// Outro exemplo

```
<?php
//Mostra estrutura do elemento XML
$file = "data.xml";
$depth = array();

function startElement($parser, $name, $attrs) {
   global $depth;
   for ($i = 0; $i < $depth[$parser]; $i++) {
       echo "  ";
   }
   echo "$name\n";
   $depth[$parser]++;
}

function endElement($parser, $name) {
   global $depth;
   $depth[$parser]--;
}
```

```
$xml_parser = xml_parser_create();
xml_set_element_handler($xml_parser, "startElement", "endElement");
if (!($fp = fopen($file, "r"))) {
   die("could not open XML input");
}

while ($data = fread($fp, 4096)) {
   if (!xml_parse($xml_parser, $data, feof($fp))) {
       die(sprintf("XML error: %s at line %d",
                   xml_error_string(xml_get_error_code($xml_parser)),
                   xml_get_current_line_number($xml_parser)));
   }
}
xml_parser_free($xml_parser);
?>
```

Arquivo a ser lido (xml_lido.php):

```
<?php
$xmlstr = <<<XML
<?xml version='1.0' encoding='ISO-8859-1' ?>
<filmes>
 <filme>
  <titulo>PHP: Iniciando o Parser</titulo>
  <personagens>
   <personagem>
   <nome>João de Brito</nome>
   <actor>Brito</actor>
   </personagem>
   <personagem>
   <nome>Manoel Cunha</nome>
   <actor>Manoel</actor>
   </personagem>
  </personagens>
  <comentario>
   O XML é uma linguagem. Ela é como uma linguagem de programação. Ou uma
   linguagem de script? Tudo será revelado após ler bem toda a
   documentação.
  </comentario>
  <votos type="thumbs">7</votos>
  <votos type="stars">5</votos>
 </filme>
</filmes>
XML;
?>
```

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/XML"
Page categories: PHP

Aplicativos em PHP/Trabalhando em PHP com/Cookies

De Wikibooks

< Aplicativos em PHP | Trabalhando em PHP com

Conteúdo

- 1 Definindo
- 2 Exemplo
- 3 Debugando Cookies
- 4 Removendo cookies
- 5 Cookies com Arrays
- 6 Mostrando o nome do visitante ao entrar com Cookies e Session

Definindo

Cookies são mecanismos para armazenar e consultar informações nos navegadores dos visitantes da página. O PHP atribui cookies utilizando a função **setcookie() ou setrawcookie()**, que deve ser utilizada antes da tag <html> numa página.

Qualquer cookie enviado por você para o cliente automaticamente será uma variável do PHP assim como dados de postagens GET ou POST, dependendo dos valores de register_globals e variables_order. Se você deseja assimilar vários valores em um único cookie, simplesmente acrescente [] ao nome do cookie.

Exemplo

```
<?php
$value = 'alguma coisa de algum lugar';

setcookie("TesteCookie", $value, time()+3600);  /* expira em 1 hora */
setcookie("TesteCookie", $value, time()+3600, "/home/ribafs/", ".exemplo.com",
1);
?>
```

Debugando Cookies

```
<?php
// Exibe um cookie individual
echo $_COOKIE["TesteCookie"];
echo $HTTP_COOKIE_VARS["TesteCookie"];

// Outra maneira de debugar/testar é visualizando todos os cookies
print_r($_COOKIE);
?>
```

Removendo cookies

```
<?php
// Ajustar para expirar em 1 hora
setcookie ("TesteCookie", "", time() - 3600);
setcookie ("TesteCookie", "", time() - 3600, "/home/ribafs/", ".exemplo.com",
1);
?>
```

Cookies com Arrays

```
<?php
// Ajustar cookies
setcookie("cookie[três]", "cookiethree");
setcookie("cookie[dois]", "cookietwo");
setcookie("cookie[um]", "cookieone");

// após a página ser carregada, mostra a saída
if (isset($_COOKIE['cookie'])) {
    foreach ($_COOKIE['cookie'] as $name => $value) {
        echo "$name : $value <br />\n";
    }
}
?>
```

Mostrando o nome do visitante ao entrar com Cookies e Session

http://www.weberforums.com/ftopic10928.html

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Trabalhando_em_PHP_com/Cookies"
Page categories: PHP

Aplicativos em PHP/Integração do PHP com SGBDs/MySQL

De Wikibooks

< Aplicativos em PHP | Integração do PHP com SGBDs

Conteúdo

- 1 Conectar ao MySQL
- 2 Fechar a Conexão
- 3 Selecionar um Banco após a Conexão
- 4 Executar Consulta SQL
- 5 Mover Ponteiro para um Registro Específico
- 6 Retornar Todos os Registos em Forma de Array
- 7 Total de Registros
- 8 Criar Banco de Dados
- 9 Nomes dos Bancos de Dados
- 10 Excluir Banco
- 11 Informações sobre os Campos
- 12 Retornar um Registro em Forma de Array de Campos
- 13 Tamanho de Campos
- 14 Nomes de Campos
- 15 Nome de Tabela
- 16 Tipos de Dados dos Campos
- 17 ID do Insert
- 18 Lista de Campos
- 19 Lista de Campos de Tabela
- 20 Lista de Tabelas de um Banco
- 21 Número de Campos
- 22 Nome de Tabela
- 23 Teste se tabela existe no MySQL
- 24 Importar e Emportar no MySQL
    - 24.1 Exportar CSV para MySQL

Conectar ao MySQL

Usaremos:

banco - cliente

```
create table clientes(id int primary key auto_increment, nome char(45), data
date);
INSERT INTO `clientes` VALUES (1, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (2, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (3, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (4, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (5, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (6, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (7, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (8, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (9, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (10, 'JoÃ£o Brito10', '2007-01-12');
INSERT INTO `clientes` VALUES (11, 'JoÃ£o Brito11', '2007-01-13');
INSERT INTO `clientes` VALUES (12, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (13, 'Jorge da Cunha', '2007-01-12');
INSERT INTO `clientes` VALUES (14, 'Jorge da Cunha', '2007-01-12');

Recomendações:

- Usar sempre os mesmos nomes de variáveis para mesmas funções. Exemplo: $link
para receber a conexÃ£o.
- Usar sempre as mesmas mensagens de erro para certos erros. Exemplo:
  mysql_query... 'Erro na consulta: ' . mysql_error() nÃ£o esquecer de passar a
funÃ£oo do MySQL que retorna o erro
```

```
$link = mysql_connect('localhost', 'root', '');
if (!$link) {
    die('Erro ao conectar conectar: ' . mysql_error());
}
```

Fechar a Conexão

```
print "<h2>FECHAR A CONEXÃO COM O MYSQL</h2>";
mysql_close($link);
```

Selecionar um Banco após a Conexão

```
print "<h2>SELECIONAR BANCO APÓS CONEXÃO</h2>";
$link = mysql_connect('localhost', 'root', '');
if (!$link) {
    die('Erro ao conectar conectar : ' . mysql_error());
}else{
        if(!$db_selected = mysql_select_db('cliente', $link))  die ('Erro ao
selecionar o banco cliente : ' . mysql_error());
}
```

Executar Consulta SQL

```
print "<h2>EXECUTAR CONSULTA(Qualquer consulta SQL)</h2>";
$result = mysql_query('SELECT * FROM clientes',$link); //se $link não for
especificado, será usado o ltimo aberto
if (!$result) {
    die('Erro na consulta: ' . mysql_error());
}
```

Mover Ponteiro para um Registro Específico

```
print "<h2>MOVER PARA UM REGISTRO ESPECÍFICO</h2>";
$query = 'SELECT nome, data FROM clientes';
$result = mysql_query($query,$link);
if (!$result) {
    die('Erro na consulta: ' . mysql_error());
}
/* obter as linhas (registros) em ordem reversa */
for ($i = mysql_num_rows($result) - 1; $i <= 0; $i--) {
    if (!mysql_data_seek($result, $i)) {
        echo "Não foi possível mover para a linha $i: " . mysql_error() . "\n";
        continue;
    }

    if (!($row = mysql_fetch_assoc($result))) {
        continue;
    }

    echo $row['nome'] . ' ' . $row['data'] . "<br />\n";
}
//mysql_free_result($result);
```

Retornar Todos os Registos em Forma de Array

```
print "<h2>RETORNAR TODOS OS REGISTROS COMO ARRAY</h2>";
mysql_fetch_array($result, MYSQL_BOTH); // Retorna todos os registros como array
numÃ©rico e nomes

$result = mysql_query("SELECT id, nome FROM clientes");

while ($row = mysql_fetch_array($result, MYSQL_BOTH)) {
    printf ("ID: %s  Nome: %s", $row[0], $row["nome"].'<br>');
}
```

Total de Registros

```
print "<h2>TOTAL DE REGISTROS</h2>";
$result = mysql_query("SELECT * FROM clientes", $link);
$num_rows = mysql_num_rows($result);

echo "$num_rows registros<br>";


<h2>Receber Registros Afetados por Consulta</h2>

<pre>
print "<h2>REGISTROS AFETADOS (INSERT, DELETE E UPDATE)</h2>";
mysql_query('DELETE FROM clientes WHERE id < 10');
printf("Registros excluídos: %d\n", mysql_affected_rows());

/* Com uma cláusula WHERE que nunca é verdadeira isso deve retornar 0 */
mysql_query('DELETE FROM mytable WHERE 0');
printf("Registros excluídos: %d\n", mysql_affected_rows());
```

Criar Banco de Dados

```
print "<h2>CRIAR BANCO DE DADOS</h2>";
$sql = 'CREATE DATABASE cliente2';
if (mysql_query($sql, $link)) {
    echo "O banco de dados cliente2 foi criado<br>";
} else {
    echo 'Erro criando o banco de dados: ' . mysql_error() . "<br>";
}
```

Nomes dos Bancos de Dados

```
print "<h2>NOMES DOS BD</h2>";
$db_list = mysql_list_dbs($link);
```

```
$i = 0;
$cnt = mysql_num_rows($db_list);
while ($i < $cnt) {
    echo mysql_db_name($db_list, $i) . "<br>";
    $i++;
}
```

Excluir Banco

```
print "<h2>EXCLUIR BANCO</h2>";
$sql = 'DROP DATABASE cliente2';
if (mysql_query($sql, $link)) {
    echo "O banco de dados foi excluÃdo com sucesso<br>";
} else {
    echo 'Erro ao excluir o banco de dados: ' . mysql_error() . "<br>";
}
<pre>


<h2>Retornar Número do Erro</h2>

<pre>
print "<h2>MYSQL_ERRNO</h2>";
if (!mysql_select_db("naoexistentebd", $link)) {
    echo mysql_errno($link) . ": " . mysql_error($link). "<br>";
}
```

Informações sobre os Campos

```
print "<h2>INFORMAÃÇÕES SOBRE CAMPOS</h2>";
$i = 0;
while ($i < mysql_num_fields($result)) {
    echo "InformaÃ§Ã£o para a coluna $i:<br><br>";
    $meta = mysql_fetch_field($result, $i);
    if (!$meta) {
        echo "Sem informaÃ§Ã£o disponÃvel<br>";
    }
    echo "<pre>
blob:         $meta->blob
max_length:   $meta->max_length
multiple_key: $meta->multiple_key
name:         $meta->name
not_null:     $meta->not_null
numeric:      $meta->numeric
primary_key:  $meta->primary_key
table:        $meta->table
type:         $meta->type
default:      $meta->def
unique_key:   $meta->unique_key
unsigned:     $meta->unsigned
zerofill:     $meta->zerofill

";
```

```
    $i++;

}
```

Retornar um Registro em Forma de Array de Campos

```
print "<h2>RETORNAR UM REGISTRO COM ARRAY DE CAMPOS</h2>";
$result = mysql_query("SELECT id,nome FROM clientes WHERE id = '12'");
if (!$result) {
    echo 'Erro na consulta: ' . mysql_error();
    exit;
}
$row = mysql_fetch_row($result);

echo $row[0]; // 42
echo $row[1]; // o valor do email
```

Tamanho de Campos

```
print "<h2>TAMANHO DE CAMPOS</h2>";
$result = mysql_query("SELECT id,nome FROM clientes WHERE id = '42'");
if (!$result) {
    echo 'Erro na consulta: ' . mysql_error();
    exit;
}

// Receberão tamanho do campo id como especificado no banco
// schema.
$length = mysql_field_len($result, 0);
echo $length;
```

Nomes de Campos

```
print "<h2>NOMES DE CAMPOS</h2>";
$res = mysql_query('select * from clientes', $link);

echo mysql_field_name($res, 0) . "<br>";
echo mysql_field_name($res, 2);
```

Nome de Tabela

```
print "<h2>NOME DE TABELA</h2>";
$table = mysql_field_table($result, $nomedeumcampodatabela);
echo $table; // people
```

Tipos de Dados dos Campos

```
print "<h2>TIPO DE DADOS DE CAMPO</h2>";
$result = mysql_query("SELECT * FROM clientes");
$fields = mysql_num_fields($result);
$rows   = mysql_num_rows($result);
$table  = mysql_field_table($result, 0);
echo "Sua tabela '" . $table . "' tem " . $fields . " campos e " . $rows . "
registros<br>";
echo "A tabela tem os seguintes campos:<br>";
for ($i=0; $i < $fields; $i++) {
    $name  = mysql_field_name($result, $i);
    $type  = mysql_field_type($result, $i);
    $len   = mysql_field_len($result, $i);
    $flags = mysql_field_flags($result, $i);
    echo $name . " " . $type . "  " . $len . " " . $flags . "<br>";
}
```

ID do Insert

```
print "<h2>RECEBENDO ID GERADO POR INSERT</h2>";
mysql_query("INSERT INTO clientes (nome) values ('Jorge da Cunha')");
printf("O ltimo registro inserido tem id %d<br>", mysql_insert_id());
```

Lista de Campos

```
print "<h2>LISTAR BANCOS DO MYSQL</h2>";
$db_list = mysql_list_dbs($link);

while ($row = mysql_fetch_object($db_list)) {
     echo $row->Database . "<br>";
}
```

Lista de Campos de Tabela

```
print "<h2>LISTAR CAMPOS DE TABELA</h2>";
while ($row = mysql_fetch_assoc($result)) {
    echo $row["id"].'-';
    echo $row["nome"].'-';
    echo $row["data"]."<br>";
}
```

Lista de Tabelas de um Banco

```
print "<h2>LISTAR TABELAS DE BANCO</h2>";
$sql = "SHOW TABLES FROM cliente";
$result = mysql_query($sql);

if (!$result) {
    echo "Erro no banco, nÃ£o pode listas as tabelas<br>";
    echo 'Erro no MySQL: ' . mysql_error();
    exit;
}

while ($row = mysql_fetch_row($result)) {
    echo "Tabela: {$row[0]}<br>";
}
```

Número de Campos

```
print "<h2>NÚMERO DE CAMPOS</h2>";
$result = mysql_query("SELECT id,nome FROM clientes WHERE id = '42'");
if (!$result) {
    echo 'Erro na consulta: ' . mysql_error();
    exit;
}

/* retorna 2 porque id,nome === 2 campos */
echo mysql_num_fields($result);
```

Nome de Tabela

```
print "<h2>NOME DE TABELA</h2>";
mysql_connect("localhost", "root", "");
$result = mysql_list_tables("cliente");

for ($i = 0; $i < mysql_num_rows($result); $i++) {
    echo "Tabela: ", mysql_tablename($result, $i), "<br>";
}


<h2>Automaticamente instalar uma tabela no MySQL com PHP</h2>

<pre>
$table_def = "id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,";
$table_def .= "student_id INT(11) NOT NULL,";
$table_def .= "f_name TINYTEXT NOT NULL,";
$table_def .= "l_name TINYTEXT NOT NULL,";
$table_def .= "supervisor TINYTEXT NOT NULL,";
$table_def .= "building TINYTEXT NOT NULL,";
$table_def .= "email TINYTEXT NOT NULL,";
$table_def .= "score SMALLINT(6) NULL,";
$table_def .= "stamp DATETIME NOT NULL,";
```

```
$table_def .= "UNIQUE KEY id (id)";

if (!@mysql_query ("CREATE TABLE $tablename ($table_def)")) {
echo "The database table, '$tablename', could not be created.";
} else {
echo "Successfully created the '$tablename' table.";
}
```

Teste se tabela existe no MySQL

```
function table_exists ($table, $db) {
        $tables = mysql_list_tables ($db);
        while (list ($temp) = mysql_fetch_array ($tables)) {
                if ($temp == $table) {
                        return TRUE;
                }
        }
        return FALSE;
}

/** How to use it **/
if (table_exists(test_table, my_database)) {
        echo"Yes the table is there.";
}
/*
akxter, http://www.oxyscripts.com/itemdisplay.php?id=1003&code=yes

And a shorter way. In the above example mysql_list_tables is deprecated in favor
of mysql_query().
*/

// here is a much more elegant method to check if a table exists ( no error
generate)

if( mysql_num_rows( mysql_query("SHOW TABLES LIKE '".$table."'")))
{
//...
}
```

Importar e Emportar no MySQL

Exportar CSV para MySQL

```
function exportarCSV_a_mySQL($fileCSV)
{
        $registros=0;

        $ruta=$fileCSV['tmp_name'];

        if(!file_exists($ruta))
        {return false;}

        $tabla=quitar_extension($fileCSV['name']);
```

```
        $borra_tabla="DROP TABLE `".$tabla."`";
        mysql_query($borra_tabla);
        $f=fopen($ruta,"r");
        if($f)
        {
                echo "<b>Guardando CSV en la BDD :</b><br />";
                $contenido=fread($f,filesize($ruta));
                fclose($f);
                $contenido=ereg_replace("\r\n", "\n" , $contenido); //
convertimos windows a unix
                $lineas=explode("\n",$contenido);
                $titulo=explode(";",$lineas[0]);
                $NUM_CAMPOS=count($titulo);
                $sql_generado_para_eliminar="";
                $crear_tabla_campos="";
                for($i=0;$i<$NUM_CAMPOS;$i++)
                {
                $titulo[$i]=ereg_replace("\"", "" , $titulo[$i]); // kitamos
comillas
                $sql_generado_para_eliminar.=" AND `".$titulo[$i]."` ='''";
                $crear_tabla_campos.="`".$titulo[$i]."` varchar(60) NOT NULL";
                        if($i+1!=$NUM_CAMPOS)// si no es el ultimo , ponemos
coma
                        {
                        $crear_tabla_campos.=",";
                        }
                }
                $crear_tabla="CREATE TABLE `".$tabla."`
(".$crear_tabla_campos.") ENGINE=MyISAM DEFAULT CHARSET=latin1;";
                mysql_query($crear_tabla);
                $linea=1;
                do
                {
                        $insertar_titulos="";
                        $insertar_campos="";
                        $campo=explode(";",$lineas[$linea]);
                        for($i=0;$i<$NUM_CAMPOS;$i++)
                        {
                        $campo[$i]=ereg_replace("\"", "" , $campo[$i]);
                        $insertar_titulos.=" `".$titulo[$i]."` ";
                        $insertar_campos.=" '".$campo[$i]."' ";
                                if($i+1!=$NUM_CAMPOS)// si no es el ultimo ,
ponemos coma
                                {
                                $insertar_titulos.=",";
                                $insertar_campos.=",";
                                }
                        }
                        $sql="INSERT INTO `".$tabla."` ( ".$insertar_titulos." )
VALUES ( ".$insertar_campos." );";
                        if(mysql_query($sql))
                        {
                        echo ". ";
                        $registros++;
                        }
                        else
                        {echo "X ";return false;}
                $linea++;
```

```
                }while(next($lineas));

        $sql="DELETE FROM `".$tabla."` WHERE
1".$sql_generado_para_eliminar;mysql_query($sql);
        echo "<br />";
        return $tabla;
        }
        else
        {
        return false;
        }
}

function quitar_extension($archivo)
{
        $extension = strrchr($archivo,".");
        $pos=strpos($archivo,$extension);
        return substr($archivo,0,$pos);
}
```

Exemplo de chamada:

```
$tabla = exportarCSV_a_mySQL($_FILES['archivo_csv']);
if($tabla)
{
echo "Export OK in mysql table : ".$tabla;
}
else
{
echo "Error in export ...";
}
```

Fonte: http://snippets.dzone.com/posts/show/4344

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Integra%C3%A7%C3%A3o_do_PHP_com_SGBDs/MySQL"
Page categories: PHP

Aplicativos em PHP/Integração do PHP com SGBDs/PostgreSQL

De Wikibooks

< Aplicativos em PHP | Integração do PHP com SGBDs

Conteúdo

- 1 Conectar ao Banco do SGBD PostgreSQL
- 2 Fechar a Conexão
- 3 Executar Consulta SQL
- 4 Encontrar Registros Afetados
- 5 Retornar Nome do Banco Conectado
- 6 Retornar um registro em forma de array de campos
- 7 Receber 1 registro na forma de array
- 8 Número de Campo no Array de Campos de um Registro
- 9 Tamanho Interno de Campo
- 10 Tipo de Dados de Campo
- 11 Última Mensagem de Erro da Conexão
- 12 Meta Dados de Tabela
- 13 Número de Campos de Consulta
- 14 Número de Registros
- 15 Codificação do Cliente
- 16 Alterar Codificação do Cliente
- 17 Versão do PostgreSQL

Conectar ao Banco do SGBD PostgreSQL

Usaremos:

banco - cliente

```
create table clientes(id serial primary key, nome char(45), data date);
INSERT INTO clientes VALUES (1, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (2, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (3, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (4, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (5, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (6, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (7, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (8, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (9, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (10, 'João Brito10', '2007-01-12');
INSERT INTO clientes VALUES (11, 'João Brito11', '2007-01-13');
INSERT INTO clientes VALUES (12, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (13, 'Jorge da Cunha', '2007-01-12');
INSERT INTO clientes VALUES (14, 'Jorge da Cunha', '2007-01-12');

Recomendações:

- Usar sempre os mesmos nomes de variáveis para mesmas funções. Exemplo: $dbcon
para receber a conexão.
- Usar sempre as mesmas mensagens de erro para certos erros. Exemplo:
  pg_query... 'Erro na consulta: ' . pg_last_error() não esquecer de passar a
função do MySQL que retorna o erro



<?php
print "<h2>CONECTAR AO BANCO NO POSTGRESQL</h2>";
$bdcon = pg_connect("dbname=cliente");
//conecta a um banco de dados chamado "cliente"

$con_string = "host=localhost port=5432 dbname=cliente user=postgres
password=postgres";
if(!$dbcon = pg_connect($con_string)) die ("Erro ao conectar ao
banco<br>".pg_last_error($dbcon));
//coneta a um banco de dados chamado "cliente" na máquina "localhost" com um
usuário e senha
```

Fechar a Conexão

```
print "<h2>
== FECHA ==
R CONEXÃO COM O BANCO</H2>";
//pg_close($dbcon)
```

Executar Consulta SQL

```
print "<h2>EXECUTAR CONSULTA (QUALQUER CONSULTA SQL)</h2>";
$result = pg_query($dbcon, "SELECT id, nome FROM clientes");
if (!$result) {
  echo "Erro na consulta.<br>";
  exit;
}

while ($row = pg_fetch_row($result)) {
  echo "ID: $row[0]  Nome: $row[1]";
  echo "<br />\n";
}
```

Encontrar Registros Afetados

```
print "<h2>NÚMERO DE REGISTROS AFETADOS (INSERT,DELETE E UPDATE)</h2>";
$result = pg_query($dbcon, "INSERT INTO clientes (id, nome,data) VALUES (40,
'João Brito', '2007-03-25')");
$cmdtuples = pg_affected_rows($result);
if ($cmdtuples==0){
        echo "Nenhum registro foi afetado.<br>";
}elseif($cmdtuples==1){
        echo $cmdtuples . " registro afetado.<br>";
}else{
        echo $cmdtuples . " registros foram afetados.<br>";
}
<pre>


<h2>Inserir Registros de Array em Tabela</h2>

<pre>
print "<h2>INSERIR REGISTROS DE ARRAY EM TABELA</h2>";
   //$db = pg_connect("dbname=cliente") or die("Erro ao conectar ao banco");
   $rows = pg_copy_to($dbcon, 'clientes');
   pg_query($dbcon, "DELETE FROM clientes");
   pg_copy_from($dbcon, 'clientes', $rows);
```

Retornar Nome do Banco Conectado

```
print "<h2>RETORNAR NOME DO BANCO CONECTADO</h2>";
  echo pg_dbname(); // mary

   if ($dbcon) {
       print "<br>Conectado com sucesso ao banco: " . pg_dbname($dbcon) .
         " em " .  pg_host($dbcon) . "<br/>\n";
   } else {
       print pg_last_error($dbcon);
       exit;
   }
```

Retornar um registro em forma de array de campos

```
print "<h2>RETORNAR UM REGISTRO COM CAMPOS EM ARRAY</h2>";

while ($row=pg_fetch_row($result,$i)) {
  for($j=0; $j < count($row); $j++) {
     echo $row[$j] . "<BR>;nbsp;";
  }
```

Receber 1 registro na forma de array

```
print "<h2>RECEBER 1 REGISTRO COMO ARRAY</h2>";
$result = pg_query($dbcon, "SELECT id, nome FROM clientes");
if (!$result) {
  echo "Erro na consulta.\n";
  exit;
}

$arr = pg_fetch_array($result, 0, PGSQL_NUM);
echo $arr[0] . " <- Registro 1 ID<br>";
echo $arr[1] . " <- Registro 1 Nome<br>";

print "<h2>NOME DE CAMPO</h2>";
  $res = pg_query($dbcon, "select * from clientes where id = '23'");
  $i = pg_num_fields($res);
  for ($j = 0; $j < $i; $j++) {
     echo "campo $j<br>";
     $fieldname = pg_field_name($res, $j);
     echo "nome do campo: $fieldname<br>";
     echo "tamanho na tela: " . pg_field_prtlen($res, $fieldname) . "
caracteres<br>";
     echo "tamanho no banco: " . pg_field_size($res, $j) . " bytes<br>";
     echo "tipo de dados: " . pg_field_type($res, $j) . " <br><br>";
  }
```

Número de Campo no Array de Campos de um Registro

```
print "<h2>NÚMERO DE CAMPO NO ARRAY DE CAMPOS DE UM REGISTRO</h2>";
  $res = pg_query($dbcon, "select id,nome,data from clientes where id = '18'");
  echo "Campo 'nome' é o campo número: ", pg_field_num($res, 'nome');
```

Tamanho Interno de Campo

```
print "<h2>TAMANHO INTERNO DE CAMPO</h2>";
  $res = pg_query($dbcon, "select * from clientes where id = '23'");
  $i = pg_num_fields($res);
  for ($j = 0; $j < $i; $j++) {
     echo "campo $j<br>";
```

```
     $fieldname = pg_field_name($res, $j);
     echo "nome do campo: $fieldname<br>";
     echo "tamanho na tela: " . pg_field_prtlen($res, $fieldname) . "
caracteres<br>";
     echo "tamanho no banco: " . pg_field_size($res, $j) . " bytes<br>";
     echo "tipo de dados: " . pg_field_type($res, $j) . " <br><br>";
  }
```

Tipo de Dados de Campo

```
print "<h2>TIPO DE DADOS DE CAMPO</h2>";
  // Assume 'title' is a varchar type
  $res = pg_query($dbcon, "select nome from clientes where id = '11'");
  echo "Tipo do campo nome: ", pg_field_type($res, 0);
```

Última Mensagem de Erro da Conexão

```
print "<h2>ÚLTIMA MENSAGEM DE ERRO DA CONEXÃO</h2>";
  // Query that fails
  $res = pg_query($dbcon, "select * from tabelaquenaoexiste");
  echo pg_last_error($dbcon);
```

Meta Dados de Tabela

```
print "<h2>META DADOS DE TABELA (EXPERIMENTAL)</h2>";
  $meta = pg_meta_data($dbcon, 'clientes');
  if (is_array($meta)) {
     echo '<pre>';
     var_dump($meta);
     echo '

';
  }
```

Número de Campos de Consulta

```
print "<h2>NÚMERO DE CAMPOS</h2>";
$num = pg_num_fields($result);
echo $num . " campos(s) retornados.<br>";
```

Número de Registros

```
print "<h2>NÚMERO DE REGISTROS</h2>";
$rows = pg_num_rows($result);
echo $rows . " registro(s) retornados.<br>";
```

Codificação do Cliente

```
print "<h2>RETORNAR CODIFICAÇÃO DO CLIENTE</h2>";
$encoding = pg_client_encoding($dbcon);
echo "A codificação do cliente é: ", $encoding, "<br>";
```

Alterar Codificação do Cliente

```
print "<h2>ALTERAR CODIFICAÇÃO DO CLIENTE</h2>";
//$dbcon = pg_pconnect("dbname=cliente");
if (!$dbcon) {
  echo "Erro na conexão.<br>";
  exit;
}

// Configurar a codificação do cliente para UNICODE. Dados devem ser
automaticamente convertidos
// da codificação do backend para a do frontend.
/* Valores possíveis SQL_ASCII, EUC_JP, EUC_CN, EUC_KR, EUC_TW, UNICODE,
MULE_INTERNAL,
LATINX (X=1...9), KOI8, WIN, ALT, SJIS, BIG5, WIN1250. */
pg_set_client_encoding($dbcon, LATIN1);
$result = pg_query($dbcon, "SELECT id,nome FROM clientes");
if (!$result) {
  echo "Erro na consulta.<br>";
  exit;
}

// Escrever dados em UTF-8
while ($row = pg_fetch_row($result)) {
  echo "ID: $row[0]  Nome: $row[1]";
  echo "<br/>";
}
```

Versão do PostgreSQL

```
print "<h2>VERSÃO DO POSTGRESQL</h2>";
print "Retorna um array com versão do cliente, protocolo e servidor (quando
disponíveis)<br>";

  $v = pg_version($dbcon);
  echo $v['client'];
```

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Integra%C3%A7%C3%A3o_do_PHP_com_SGBDs/PostgreSQL"
Page categories: PHP

Aplicativos em PHP/Integração do PHP com SGBDs/SQLite

De Wikibooks

< Aplicativos em PHP | Integração do PHP com SGBDs

12.3 - Integração do PHP com SQLite

O SQLite é um SGBD leve que pode ser utilizado integrado ao PHP para pequenos e médios aplicativos.

Aqui encontrará um aplicativo com o básico CRUD em PHP com acesso ao SQLite:

http://www.ribafs.net/down/appsexemplo/sqlite/iniciante_sqlitephp.zip

SQLite and PHP

http://www.php.net/~wez/SQLite_and_PHP.ppt

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Integra%C3%A7%C3%A3o_do_PHP_com_SGBDs/SQLite"
Page categories: PHP

Aplicativos em PHP/Integração do PHP com SGBDs/IBM DB2 Express

De Wikibooks

< Aplicativos em PHP | Integração do PHP com SGBDs

```
  Configuração utilizada no sistema operacional Windows XP SP2:
```

-Baixe a extensão (pecl5.2-win32-latest) do IBM DB2 para o PHP em:

-(http://pecl.php.net/package/PDO/0.3)

-(http://snaps.php.net/win32/)

- Instale a mesma (php_ibm_db2.dll) na pasta do PHP (ex:C:\web\php-5.1.4-Win32\ext);

- Edite as linhas do php.ini identificando o diretório dos arquivos de extensão

(ex: extension_dir = "c:/web/php-5.1.4-Win32/ext/") e habilite a linha que refere-se ao db2

(extension=php_ibm_db2.dll) retirando o comentário(;);

- Depois é só aproveitar os recursos do PHP e DB2:

- Exemplo de conexão (conecta.php):

```
<?php
function dbconnect($verbose) {
    $dbname = "Nome_Banco";
    $username = "Usuario_de_Conexao";
    $password = "Senha";

    $dbconn = odbc_connect($dbname, $username, $password);

    if (($verbose == TRUE) && ($dbconn == 0)) {
        $sqlerror = odbc_errormsg($dbconn);
    }
    return($dbconn);
   }
?>
```

- Exemplo de Inserção:

```
<?php
include("conecta.php");

  $sql = "INSERT INTO estado VALUES(3,'Rio Grande do Sul')";

  $verbose = TRUE;
  $dbconn = dbconnect($verbose);

  if ($dbconn != 0) {
```

```
     $result = odbc_exec($dbconn, $sql);

     if ($result == 0) {
        echo("INSERT Falhou !!");
        $sqlerror = odbc_errormsg($dbconn);
        echo($sqlerror);
     }
     else {
          echo("INSERT realizado com sucesso !!");
     }
}
?>
```

Obtido em "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Integra%C3%A7%C3%A3o_do_PHP_com_SGBDs/IBM_DB2_Express"

Aplicativos em PHP/Recursos Extras/Controle de Estoque

De Wikibooks

< Aplicativos em PHP | Recursos Extras

13.1 - Controle de Estoque

Três sistemas de exemplo de ontrole de Estoque em:

http://ribafs.net - Seção Aplicativos de Exemplo - MySQL

Exemplo de controle de Estoque no Viva o Linux - http://www.vivaolinux.com.br/scripts/verScript.php?codigo=694

LinuxStok - Gestão Comercial - http://linuxstok.wikidot.com/

Stoq - Gestão Comercial - Instalação e configuração - http://www.vivaolinux.com.br/artigos/impressora.php?codigo=6767

Implantação de Controle de Estoque em uma Clínica Odontológica - http://www.em.ufop.br/depro/curso/monografias/2003renata.pdf

Sistemas de Controle de Estoque no superdownloads - Não são em PHP mas ajudam a ter idéia como a coisa funciona e algumas funcionalidades que devemos implementar:

http://superdownloads.uol.com.br/busca/controle-de-estoque.html

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Controle_de_Estoque"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Loja Virtual

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Conteúdo

- 1 Loja Virtual
- 2 Carrinho de compras com PHP e MySQL
- 3 osCommerce
  - 3.1 SPS BRADESCO para OsCommerce
  - 3.2 osCommerce - Tradução OMeuNuke
    - 3.2.1 Passos para a Instalação do osCommerce
    - 3.2.2 Toturial de Instalação do Boleto Bancário (Adaptação do site do O Meu Nuke)
  - 3.3 osCommerce - Tradução PHPMania
- 4 VirtueMart

Loja Virtual

Carrinho de compras com PHP e MySQL

http://www.revistaphp.com.br/print.php?id=150

Arquivos - http://www.revistaphp.com.br/abrecodigos.php?id=27

osCommerce

No Brasil temos atualmente duas principais traduções do osCommerce, que devemos preferir à versão em inglês pelas facilidades adicionadas, como suporte ao nosso idioma ativado, como suporte aos nossos tipos de pagamentos: boletos bancários, cartões de crédito e outros.

SPS BRADESCO para OsCommerce

Comentário no PHPBrasil:

Licença: GNU General Public License

Descrição: Pessoal, segue modulo SPS BRADESCO.

Este modulo tem como função realizar a integração com as formas de pagamento Bradesco, com uma loja OSCOMMERCE.

Os modulos disponiveis são, BOLETO COM RETORNO, FINANCIAMENTO, CARTÃO e TRANSFERENCIA ENTRE CONTAS.

Tudo começa na pagina spsbradesco_metodospagamento.php

AJUSTE NO CODIGO, COLOCANDO SUAS INFORMAÇÕES. <<informe o MerchantId>> <<informe numero conta>> <<INFORME NUMERO AGENCIA>> <<INFORME CEDENTE>>

Como tinha dito esse metodo foi desenvolvido as pressas, tem muita referencia duplicada e muita coisa e fora do padrão OSC.

Estou a disposição para esclarecer qualquer duvida.

Se alguem gostou e quiser fazer uma contribuição segue link.
http://www.bitebitsolucoes.com/contribuicao.php

Atencioamente,

BiteBit Soluções http://www.bitebitsolucoes.com

http://phpbrasil.com/scripts/script.php/id/3665

Download - http://phpbrasil.com/scripts/download.php/id/5590

osCommerce - Tradução OMeuNuke

ALERTA: Após instalar o oscommerce, adicione autenticação ao script admin/index.php, pois o mesmo está aberto a qualquer visitante sem qualquer autenticação ou então use os recursos do seu servidor de hospedagem para isso.

Site oficial - http://www.oscommerce.com/

Download da versão traduzida pelo pessoal do OmeuNuke - http://www.omeunuke.com/modules.php?name=Downloads&op=getit&lid=22

Requisitos do osCommerce:

- MySQL versão 4.x ou superior.

- PHP4 e PHP5.

- register_globals = On (no php.ini)

Adicionais - http://www.omeunuke.com/modules.php?name=Downloads&cid=5

Módulos de Pagamento (boleto, transferência bancária, BRPay e contra entrega) - http://www.omeunuke.com/modules.php?name=Downloads&cid=8

Templates - http://www.omeunuke.com/modules.php?name=Downloads&cid=9

Tutoriais:

Instalaço do osCommerce

http://www.omeunuke.com/modules.php?name=Sections&op=printpage&artid=29

Outros tutoriais: http://www.omeunuke.com/modules.php?name=Sections&op=listarticles&secid=5

Forums - http://www.omeunuke.com/modules.php?name=Forums

Obs.: Sempre ao acessar cada uma dessas páginas role a tela até o final da mesma.

Obs2.: Faça seu registro no site omeunuke para ter acesso ao conteúdo e arquivos.

Passos para a Instalação do osCommerce

Habilitar (On) "register_globals" no php.ini.

No Xampp para php4 ou 5 o php.ini fica em apache\bin

Descompactar em oscommerce e chamar com:

http://localhost/oscommerce/install

- Clicar em Install

- Deixe os dois checks marcados e clique em Continue

- Entrar com os dados do banco de dados e clique em Continue. O usuário deve ter poder de criar o banco ou então cria você mesmo

- Clique em Continue se tudo bem

- Em database import clique em Continue

- Na tela Configuration apenas Continue

- Continue novamente para confirmar

- Agora recebemos dois botões: Catalog (loja) e Administration Tool (painel de controle administrativo)

Toturial de Instalação do Boleto Bancário (Adaptação do site do O Meu Nuke)

Instruções para instalação do modulo de boleto bancário - Por Marcelo73

Adaptações - Ribamar FS

Baseado no tutorial de EOF

Baixe os arquivos AQUI - http://www.omeunuke.com/modules.php?name=Downloads&op=getit&lid=83

Deve baixar o arquivo Boletobancario.zip e descompactar em uma pasta temporária.

INSTRUÇÕES

Após descompactar terá:

boletophp (pasta)

Modulo para oscommerce (pasta)

leia.txt (arquivo)

Apenas trabalharemos com os arquivos da pasta Modulo para oscommerce

Acesse a pasta "Modulo para oscommerce" e veja que tem dois diretórios: boleto e catalog.

1. Agora acesse a pasta "catalog" e verá uma estrutura assim

```
includes
    languages
        portuguese
            modules
                payment
                    boletophp.php (copie este arquivo para
                        c:\xampp\htdocs\oscommerce\includes\languages\portuguese
\modules\payment)
    modules
        payment
            boletophp.php (copie este arquivo para
c:\xampp\htdocs\oscommerce\includes\modules\payment)
```

2. Acesse o Módulo de Pagamentos no painel de controle de sua loja virtual e clique em instalar

CONFIGURAÇÃO DO MODULO

Geração de Boletos Bancários utilizando o BoletoPHP

www.boletophp.com.br

Gerar Boletos Bancários - BoletoPHP

True

URL do boleto no Sistema BoletoPHP

UTILIZE O CAMINHO COMPLETO DE ONDE SE ENCONTRA O SEU ARQUIVO PARA GERAÇÃO DO BOLETO BANCÁRIO,BOLETO_NOMEDOBANCO.PHP ONDE NOME DO BANCO É O NOME DO ARQUIVO QUE VOCÊ ESTA INSTALANDO PARA A GERAÇÃO DO BOLETO.

http://www.lojavirtual.com.br/boleto/boleto_bb.php

Prazo de Vencimento

5

Demonstrativo

Pagamento de Compra na Loja Virtual http://www.lojavirtual.com.br

Instruções - Linha 1

Sr. Caixa , não receber após a Data de Vencimento.

Instruções - Linha 2

- Em caso de dúvidas entre em contato conosco: vendas@lojavirtual.com.br

Zonas suportadas

--none--

Status dos pedidos

Pending

Ordem de exibição

0

3. Dentro do seu arquivo de configuração do banco que você vai usar do BoletoPhp

(Exemplo Banco do Brasil: .../boleto/boleto_bb.php)

Faça as seguintes substituições:

```
// DADOS DO BOLETO PARA O SEU CLIENTE
$taxa_boleto = 2.95; // Valor da Taxa do Boleto
$valor_cobrado = $ValorDocumento;
$valor_cobrado = str_replace(",", ".",$valor_cobrado);
$valor_boleto=number_format($valor_cobrado+$taxa_boleto, 2, ',', ''); // Valor
que será cobrado do seu cliente com a Taxa do Boleto

$dadosboleto["inicio_nosso_numero"] = "80"; // Inicio do Nosso numero - Pode ser
80 ou 81 ou 82 (Somente se for Caixa Economica)
$dadosboleto["nosso_numero"] = $NumeroDocumento; // Nosso numero - REGRA:
Verifique o Máximo permitido pelo seu banco!
$dadosboleto["numero_documento"] = $NumeroDocumento; // Num do pedido ou do
documento
$dadosboleto["data_vencimento"] = $DataVencimento; // Data de Vencimento do
Boleto - REGRA: Formato DD/MM/AAAA
$dadosboleto["data_documento"] = date("d/m/Y"); // Data de emissão do Boleto
$dadosboleto["data_processamento"] = ""; // Data de processamento do boleto
```

```
(opcional)
$dadosboleto["valor_boleto"] = $valor_boleto; // Valor do Boleto - REGRA: Com
vírgula e sempre com duas casas depois da virgula

// DADOS DO SEU CLIENTE
$dadosboleto["sacado"] = $NomeSacado;
$dadosboleto["endereco1"] = $EnderecoSacado." - ".$BairroSacado;
$dadosboleto["endereco2"] = $CidadeSacado." - ".$EstadoSacado." - CEP:
".$CepSacado;

// INFORMACOES PARA O CLIENTE
$dadosboleto["demonstrativo"] = $Demonstrativo;
$dadosboleto["instrucoes"] = "Valor referente a Compra Nr.".$NumeroDocumento."
Taxa bancária - R$ ".$taxa_boleto;
$dadosboleto["instrucoes1"] = "Joja Virtual do Curso de PHP -
http://www.lojavirtual.com.br";
$dadosboleto["instrucoes2"] = $InstrucoesCaixaCedente;
$dadosboleto["instrucoes3"] = $InstrucoesCaixaCedente2;
$dadosboleto["instrucoes4"] = "";
```

4. Copie a pasta boleto para o raiz do oscommerce que instalou

Exemplo, copie para cá c:\xampp\htdocs\oscommerce.

5. Faça uma compra veja se aparecerá o Boleto como opçao de pagamento

Imprima o boleto e simule o pagamento deste boleto junto ao seu banco na internet para ver se os dados estão corretos

Este artigo veio de OmeuNuke

http://www.omeunuke.com

osCommerce - Tradução PHPMania

Site - http://www.phpmania.org/

Download da Loja, de Módulos de Pagemento e outros - http://www.phpmania.org/modules.php?name=Downloads&d_op=viewdownload&cid=2

VirtueMart

Este funciona com o Joomla. Existe uma versão do Joomla apenas com o a loja virtual VirtueMart:

http://virtuemart.net/index.php?option=com_docman&task=cat_view&gid=101&Itemid=66

Basta instalar em uma instalação tipo a do Joomla.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Loja_Virtual"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Código de Barras

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Conteúdo

- 1 Código de Barras em PHP
- 2 Código de barras 2 de 5 intercalado
- 3 Exemplo de Código de Barras em PHP - do Sidiney Sodré Gali de Souza
- 4 Outro exemplo, este do Marcolino
- 5 Pear parte 5 - Gerando códigos de barras padrão 128

Código de Barras em PHP

Código de barras 2 de 5 intercalado

Script no PHPBrasil:

http://phpbrasil.com/scripts/script.php/id/3663

Exemplo de Código de Barras em PHP - do Sidiney Sodré Gali de Souza

```
<?php
/*
Autor: Sidiney Sodré Gali de Souza <sidsodre@bol.com.br>
Data: 21/11/2005

Código de Barras - código-fonte   em:
http://www.vivaolinux.com.br/scripts/verFonte.php?codigo=1394
*/

function CodigoBarras($code) {
  $lw = 2; $hi = 100;
  $Lencode = array('0001101','0011001','0010011','0111101','0100011',
                   '0110001','0101111','0111011','0110111','0001011');
  $Rencode = array('1110010','1100110','1101100','1000010','1011100',
                   '1001110','1010000','1000100','1001000','1110100');
  $ends = '101'; $center = '01010';
  /* UPC-A Must be 11 digits, we compute the checksum. */
```

```
  if ( strlen($code) != 11 ) { die("UPC-A Must be 11 digits."); }
  /* Compute the EAN-13 Checksum digit */
  $ncode = '0'.$code;
  $even = 0; $odd = 0;
  for ($x=0;$x<12;$x++) {
   if ($x % 2) { $odd += $ncode[$x]; } else { $even += $ncode[$x]; }
  }
  $code.=(10 - (($odd * 3 + $even) % 10)) % 10;
  /* Create the bar encoding using a binary string */
  $bars=$ends;
  $bars.=$Lencode[$code[0]];
  for($x=1;$x<6;$x++) {
   $bars.=$Lencode[$code[$x]];
  }
  $bars.=$center;
  for($x=6;$x<12;$x++) {
   $bars.=$Rencode[$code[$x]];
  }
  $bars.=$ends;
  /* Generate the Barcode Image */
  $img = ImageCreate($lw*95+30,$hi+30);
  $fg = ImageColorAllocate($img, 0, 0, 0);
  $bg = ImageColorAllocate($img, 255, 255, 255);
  ImageFilledRectangle($img, 0, 0, $lw*95+30, $hi+30, $bg);
  $shift=10;
  for ($x=0;$x<strlen($bars);$x++) {
   if (($x<10) || ($x>=45 && $x<50) || ($x >=85)) { $sh=10; } else { $sh=0; }
   if ($bars[$x] == '1') { $color = $fg; } else { $color = $bg; }
   ImageFilledRectangle($img, ($x*$lw)+15,5,($x+1)*$lw+14,$hi+5+$sh,$color);
  }
  /* Add the Human Readable Label */
  ImageString($img,4,5,$hi-5,$code[0],$fg);
  for ($x=0;$x<5;$x++) {
   ImageString($img,5,$lw*(13+$x*6)+15,$hi+5,$code[$x+1],$fg);
   ImageString($img,5,$lw*(53+$x*6)+15,$hi+5,$code[$x+6],$fg);
  }
  ImageString($img,4,$lw*95+17,$hi-5,$code[11],$fg);
  /* Output the Header and Content. */
  header("Content-Type: image/png");
  ImagePNG($img);
}

CodigoBarras('54879874585');
?>
```

Outro exemplo, este do Marcolino

```
<?php

/* Geração de código de barras padrão p/ boletos bancários brasileiros.

Autor: MARCOLINO, Alexandre de Jesus ( marcolino@facil.com )

Este código pode ser usado e copiado livremente desde que site e comunique o
autor por email.
```

```
Testado sob o Internet Explorer 6 e o Netscape 7.

*/

function montacodigodebarras($valor) {
  $lw = 1 ; $hi = 50;
/* Tabela de código de barras Padrão Brasileiro para boletos bancários

  Barra valor 00 =0000111100
  Barra valor 01 =0100101001
  Barra valor 02 =0001101001
  Barra valor 03 =0101101000
  Barra valor 04 =0000111001
  Barra valor 05 =0100111000
  Barra valor 06 =0001111000
  Barra valor 07 =0000101101
  Barra valor 08 =0100101100
  Barra valor 09 =0001101100
  Barra valor 10 =1000010110
  Barra valor 11 =1100000011
  Barra valor 12 =1001000011
  Barra valor 13 =1101000010
  Barra valor 14 =1000010011
  Barra valor 15 =1100010010
  Barra valor 16 =1001010010
  Barra valor 17 =1000000111
  Barra valor 18 =1100000110
  Barra valor 19 =1001000110
  Barra valor 20 =0010010110
  Barra valor 21 =0110000011
  Barra valor 22 =0011000011
  Barra valor 23 =0111000010
  Barra valor 24 =0010010011
  Barra valor 25 =0110010010
  Barra valor 26 =0011010010
  Barra valor 27 =0010000111
  Barra valor 28 =0110000110
  Barra valor 29 =0011000110
  Barra valor 30 =1010010100
  Barra valor 31 =1110000001
  Barra valor 32 =1011000001
  Barra valor 33 =1111000000
  Barra valor 34 =1010010001
  Barra valor 35 =1110010000
  Barra valor 36 =1011010000
  Barra valor 37 =1010000101
  Barra valor 38 =1110000100
  Barra valor 39 =1011000100
  Barra valor 40 =0000110110
  Barra valor 41 =0100100011
  Barra valor 42 =0001100011
  Barra valor 43 =0101100010
  Barra valor 44 =0000110011
  Barra valor 45 =0100110010
  Barra valor 46 =0001110010
  Barra valor 47 =0000100111
  Barra valor 48 =0100100110
  Barra valor 49 =0001100110
```

```
  Barra valor 50 =1000110100
  Barra valor 51 =1100100001
  Barra valor 52 =1001100001
  Barra valor 53 =1101100000
  Barra valor 54 =1000110001
  Barra valor 55 =1100110000
  Barra valor 56 =1001110000
  Barra valor 57 =1000100101
  Barra valor 58 =1100100100
  Barra valor 59 =1001100100
  Barra valor 60 =0010110100
  Barra valor 61 =0110100001
  Barra valor 62 =0011100001
  Barra valor 63 =0111100000
  Barra valor 64 =0010110001
  Barra valor 65 =0110110000
  Barra valor 66 =0011110000
  Barra valor 67 =0010100101
  Barra valor 68 =0110100100
  Barra valor 69 =0011100100
  Barra valor 70 =0000011110
  Barra valor 71 =0100001011
  Barra valor 72 =0001001011
  Barra valor 73 =0101001010
  Barra valor 74 =0000011011
  Barra valor 75 =0100011010
  Barra valor 76 =0001011010
  Barra valor 77 =0000001111
  Barra valor 78 =0100001110
  Barra valor 79 =0001001110
  Barra valor 80 =1000011100
  Barra valor 81 =1100001001
  Barra valor 82 =1001001001
  Barra valor 83 =1101001000
  Barra valor 84 =1000011001
  Barra valor 85 =1100011000
  Barra valor 86 =1001011000
  Barra valor 87 =1000001101
  Barra valor 88 =1100001100
  Barra valor 89 =1001001100
  Barra valor 90 =0010011100
  Barra valor 91 =0110001001
  Barra valor 92 =0011001001
  Barra valor 93 =0111001000
  Barra valor 94 =0010011001
  Barra valor 95 =0110011000
  Barra valor 96 =0011011000
  Barra valor 97 =0010001101
  Barra valor 98 =0110001100
  Barra valor 99 =0011001100
*/

/* A lógica abaixo aplica a tabela acima para dentro da variável tabcodbarra{}
*/

  $tabcodbarra[0] = "00110" ;
  $tabcodbarra[1] = "10001" ;
  $tabcodbarra[2] = "01001" ;
  $tabcodbarra[3] = "11000" ;
```

```
  $tabcodbarra[4] = "00101" ;
  $tabcodbarra[5] = "10100" ;
  $tabcodbarra[6] = "01100" ;
  $tabcodbarra[7] = "00011" ;
  $tabcodbarra[8] = "10010" ;
  $tabcodbarra[9] = "01010" ;

  for($f1=9;$f1>=0;$f1--){
    for($f2=9;$f2>=0;$f2--){
      $f = ($f1 * 10) + $f2 ;
      $texto = "" ;
      for($i=1;$i<6;$i++){
        $texto .=  substr($tabcodbarra[$f1],($i-1),1) .
substr($tabcodbarra[$f2],($i-1),1);
      }
      $tabcodbarra[$f] = $texto;
   }
  }


/* Gerando a Imagen do Código de Barras */

/* Gera a base para o preenchimento do código.
$img    -> imagem do código
$branco -> cor branca
$preto  -> cor preta ( ou afrobrasileira, sem preconceito nenhum pô )
*/
      $img = ImageCreate($lw*95+1000,$hi+30);

      $preto  = ImageColorAllocate($img, 0, 0, 0);
      $branco = ImageColorAllocate($img, 255, 255, 255);

/* Cria o retángulo principal onde a imagem será aplicada. */

      ImageFilledRectangle($img, 0, 0, $lw*95+1000, $hi+30, $branco);

# Abertura do código de barras.
      ImageFilledRectangle($img, 1,5,1,65,$preto);
      ImageFilledRectangle($img, 2,5,2,65,$branco);
      ImageFilledRectangle($img, 3,5,3,65,$preto);
      ImageFilledRectangle($img, 4,5,4,65,$branco);

# Aqui começa a varredura no valor enviado para confecção do código de barras.

$fino = 1 ;
$largo = 2.72 ; // Para o windows funciona mas o tamanho ideal seria 3
$pos   = 5 ;
$texto = $valor ;
if((strlen($texto) % 2) <> 0){
        $texto = "0" . $texto;
}

// Draw dos dados
while (strlen($texto) > 0) {
  $i = round(esquerda($texto,2));
  $texto = direita($texto,strlen($texto)-2);

  $f = $tabcodbarra[$i];
```

```
  for($i=1;$i<11;$i+=2){
    if (substr($f,($i-1),1) == "0") {
      $f1 = $fino ;
    }else{
      $f1 = $largo ;
    }

/* Imprimindo uma barra preta */
  ImageFilledRectangle($img, $pos,5,$pos-1+$f1,65,$preto)  ;
  $pos = $pos + $f1 ;
/* Fim da impressão da barra preta */

  if (substr($f,$i,1) == "0") {
      $f2 = $fino ;
    }else{
      $f2 = $largo ;
    }

/* Imprimindo uma barra branca */
  ImageFilledRectangle($img, $pos,5,$pos-1+$f2,65,$branco)  ;
  $pos = $pos + $f2 ;
/* Fim da impressão da barra branca */

  }
}

# Fechamento do código de barras.

ImageFilledRectangle($img, $pos,5,$pos-1+$largo,65,$preto);
$pos=$pos+$largo;

ImageFilledRectangle($img, $pos,5,$pos-1+$fino,65,$branco);
$pos=$pos+$fino;


ImageFilledRectangle($img, $pos,5,$pos-1+$fino,65,$preto);
$pos=$pos+$fino;


/* MANDA A IMAGEM PRO BROWSER */
      header("Content-Type: image/png");
      ImagePNG($img);
}
//montacodigodebarras('237942258000000419333940600920012801000010 20');

//$valorpassado = $_SERVER['QUERY_STRING'] ;

montacodigodebarras('237912274000000419333940600920012802000010 20');

montacodigodebarras($_SERVER['QUERY_STRING']) ;

function esquerda($entra,$comp){
        return substr($entra,0,$comp);
}

function direita($entra,$comp){
        return substr($entra,strlen($entra)-$comp,$comp);
}
```

?>

Pear parte 5 - Gerando códigos de barras padrão 128

http://www.revistaphp.com.br/print.php?id=146

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/C%C3%B3digo_de_Barras"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Boleto bancário

De Wikibooks

< Aplicativos em PHP | Recursos Extras

13.4 - Boleto Bancário

PHP Boleto

O objetivo desse projeto é de criar um sistema de geração e manutenção de boletos. Ele poderá ser usado por sites de e-Commerce ou empresas em geral para simplificar o modelo de pagamento e incentivar os clientes a pagarem seus boletos pela própria Web.

http://sourceforge.net/projects/phpboleto

Contribuindo: Encontrei esse projeto e espero que seja útil. http://www.boletophp.com.br/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Boleto_banc%C3%A1rio"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/PHP-GTK

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Através da biblioteca GTK podemos utilizar o PHP para criação de aplicativos desktop.

PHP-GTK na Wikipedia

PHP-GTK é uma extensão para a linguagem de programação PHP que implementa o binding da linguagem para o GTK+. Proporciona uma interface orientada a objeto para as classes e funções do GTK+ e simplifica, e muito, o desenvolvimento de aplicações desktop multi-plataforma. Seu autor original é Andrei Zmievski e o desenvolvimento está em andamento.

Usos significantes do PHP-GTK incluem:

```
* Agata Report - Uma aplicação de geração de relatórios;

* Tulip - Um ambiente de desenvolvimento para PHP e PHP-GTK;
* LinuxStok - Controle de Estoque e Financeiro.
```

Links externos

```
* Website oficial do PHP-GTK
* Comunidade brasileira de PHP-GTK
```

Ferramenta RAD para criação de Interfaces em PHP-GTK

Disponível para Windows e Linux

Site oficial - http://anubis.php-gtk.com.br/

Anubis é um aplicativo RAD (Rapid Application Development) IDE (Integrated Development Environment) auxilia o desenvolvimento de aplicações na linguagem PHP-GTK - com ele você "desenha" as telas de seus aplicativos e o código é gerado automaticamente na linguagem - isto implica em maior rapidez na execução do programa pois não há necessidade de se interpretar arquivos externos como os do glade.

Instalação do PHP-GTK no Windows

Download - http://www.php-gtk.com.br/windows

Executar o Instalador PhpGtk2setup.exe

Veja a estrutura criada:

C:\php-gtk2

```
php.exe = Interpretador PHP5 com suporte ao GTK2;
php-gtk-brasil.exe = Nosso programa de boas-vindas;
php.ini = arquivo de configuração do PHP;
\ext = extensões do PHP;
```

As aplicações php-gtk podem ser inicializadas pela linha de comando:

C:\php-gtk2\php C:\php-gtk2\start.php

Ou sem a janela do MS-DOS:

C:\php-gtk2\php-win C:\php-gtk2\start.php

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/PHP-GTK"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Contadores de visita

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Contadores

Contador de Acesso em PHP

Do Ândres Oliveira na Revista do PHP.

http://revistaphp.com.br/artigo.php?id=139

Criamos um contador, programado em PHP, que leva a conta das impressões que foram realizadas numa página web, utilizando um arquivo de texto como apoio.

O objetivo é levar uma contagem das vezes que uma página foi visitada.

Colocamos o seguinte script PHP ao final da página, se entenderá bem se for lido o artigo citado antes.

```
<?php
  $arquivo = "contador.txt";
  $contador = 0;
  $fp = fopen($arquivo,"r");
  $contador = fgets($fp, 26);
  fclose($fp);
  ++$contador;
  $fp = fopen($arquivo,"w+");
  fwrite($fp, $contador, 26);
  fclose($fp);
  echo "Esta página foi visitada $contador vezes";
?>
```

www.revistaphp.com.br

Ademais, criamos um arquivo chamado "contador.txt" que salvamos no mesmo diretório que a página. Tal arquivo foi inicializado com um zero (0) como único texto.

Nota: Se tiver problemas na hora de escrever em um arquivo, quase com toda probabilidade, estará protegido contra escritura. Ou o arquivo ou então o diretório.

Se você tiver seu próprio servidor terá que modificar as permissões do tal arquivo ou diretório por sua conta. Entretanto, se estiver publicando em uma hospedagem contratada em uma provedor terá que se informar de que mecanismo há que colocar em funcionamento nesse provedor para conseguir as permissões. Em muitos casos existirá um painel de controle onde modificar essas coisas, em

outros casos terá que escrever ao suporte técnico para que eles façam a mão ou que digam como fazer, se é que é possível.

Com isso já está feito um contador muito simples, mas muito funcional.

Site: http://www.netcenterdesign.com.br

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Contadores_de_visita"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Livros de visita

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Livros de Visitas em PHP

Exemplo do Éder - http://www.phpbrasil.com/scripts/script.php/id/93

Livro de Visitas Prático - http://www.phpbrasil.com/scripts/script.php/id/126

Livro de Visitas Personalizável - http://www.phpbrasil.com/scripts/script.php/id/183

Livro de Visitas com Admin - http://www.phpbrasil.com/scripts/script.php/id/2557

Livro de Visitas TXT - http://www.phpbrasil.com/scripts/script.php/id/2832

Advanced GuestBook - http://www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=1362&lngWId=8

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Livros_de_visita"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Chats

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Chats em PHP

PHPOpenChat - Chat bom em PHP

Download - http://phpopenchat.sourceforge.net/

Faça o download e descompacte no diretório web em phpopenchat

Abrir no browser - http://127.0.0.1/phpopenchat/

- Clicar em use the installer - Alterar a permissão de escrita para: - config.inc.php - tmp - include/templates/openchat/images/icons/chatter

- Clique em next

A great and simple online chat

http://www.Planet-Source-Code.com/vb/scripts/ShowCode.asp?txtCodeId=1091&lngWId=8

Um Simples Chat sem Banco de Dados

http://www.phpbrasil.com/articles/article.php/id/166

Com este exstem diversos outros exemplos de chat no PHPBrasil.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Chats"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Geradores de Relatórios

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Conteúdo

- 1 BIRT - Business Intelligence and Report Tools
  - 1.1 Criando um Novo Projeto do Tipo Relatório
  - 1.2 1 - Criando um relatório partindo de um blank report design
  - 1.3 2 - Adicionar um Data Source (Fonte de dados)
  - 1.4 3 - Criar um Data Set
  - 1.5 4 - Layout do relatório
  - 1.6 5 - Ordenando registros
  - 1.7 6 - Adicionando Título ao Relatório
- 2 Utilizando o BIRT com PHP ou simplesmente via Web
  - 2.1 Tutorial de Instalação e uso do Tomcat e do BIRT
  - 2.2 Instalação e Configuração do J2SE
  - 2.3 Instalando e configurando o Tomcat
  - 2.4 Testando o Tomcat
  - 2.5 Administrando o Tomcat
  - 2.6 Instalando e usando o Visualizador de Relatórios do BIRT
  - 2.7 Instalando o Driver JDBC do seu SGBD
  - 2.8 Abrindo Relatório de Script PHP
  - 2.9 Passando parâmetros
  - 2.10 Parâmetros de formulários
  - 2.11 Opções avançadas
- 3 Gerador de Relatório iReport
  - 3.1 Ótimo tutorial sobre o iReport, inclusive com o acesso via web (com java)
  - 3.2 Tutorial de iReport
  - 3.3 Starting with JasperReports by Gregory Beumer
  - 3.4 Manual do iReport
  - 3.5 iReport Documentação
  - 3.6 Tutorial JasperReports - IReports
  - 3.7 Outro tutorial em português
  - 3.8 A Tutorial on Generating Reports by iReport
- 4 Gerando relatórios em PHP com fPDF
  - 4.1 Tutorial muito bom - PDF Generation With PHP
  - 4.2 Gerando Documentos PDF com PHP (fPDF)

BIRT - Business Intelligence and Report Tools

O BIRT é um gerador de relatórios aos moldes do Crystal Report e do iReport, que inclusive tem integração com o PHP e outras linguagens server side.

Download - http://download.eclipse.org/birt/downloads/

Criando um Novo Projeto do Tipo Relatório

- New - Project - Business Intelligence and Report Tools - Report Project

- Clicar sobre o nome do Projeto criado com o botão direito e New - Report

1 - Criando um relatório partindo de um blank report design

- Crie um novo projeto

- Crie um novo Relatório

- Em File name entre com clientes.rptdesign e Next

- Em Report template deixe Blank Report e finish

- Observe que a janela de Layout (ao centro e à direita) exibe o nome do relatório acima e seu conteúdo está vazio. À esquerda temos a Paleta de Ferramentas para o design do relatório. Logo à direita da Paleta temos a aba Data Explorer, para a conexão com bancos de dados.

2 - Adicionar um Data Source (Fonte de dados)

Do tipo JDBC do PostgreSQL, para usar um banco de dados no nosso relatório.

- Clicar na aba Data Explorer (à direita da Paleta)

- Clicar com botão direito - New DataSource

- Selecionar JDBC Data Source - Digitar um nome (Usuário) e Next

- Deve antes ter o jdbc para o postgreSQL em algun diretório

- Clicar em Manage Drivers

- Clicar em Add e indicar o driver jdbc do portgresql

- Selecionar e OK

- Em Driver Class selecionar o driver correto

- Em Database URL entre com algo como:

jdbc:postgresql://127.0.0.1:5432/db_usuario

- Digite username e password do postgresql

- Clique em Test Connection

- Se tudo ok clique em OK

3 - Criar um Data Set

- Clicar em Data Sets com o botão direito e New DataSet

- Em Data set name entre com um nome (Usuário Ceará)

- Em Data source aceite o default

- Em data set type também aceite o default e clique em Next

- Na caixa da direita clique após o select e digite "nome, email". Clique após o from e dê um duplo clique no nome da tabela. Ao invés de digitar os campos também podemos efetuar duplos clique nos mesmos.

Para ter uma consulta dinâmica adicione parâmetros assim: "select nome,email from public.usuario WHERE nome LIKE " + params["nome"]

- Clique em Finish. Apenas clique em OK.

- Duplo clique no data set clientesceara à esquerda e clique em Preview Results para visualizar os registros e se certificar de que está tudo ok.

Veja que também existe suporte a JOIN. Basta clicar com o botão direito sobre Data Sets e New Join Data Set

4 - Layout do relatório

Tabelas (tables) interagem com todos os registros retornados por um data set. Isto torna fácil a exibição de registros em forma de linhas e colunas.

- Arraste o controle Table da paleta e solte no editor de layout. Será perguntado sobre quantas colunas e quantos detalhes. Deixe com 2 e 1 e OK e feche o Data Explorer, pois usaremos somente os campos nome e email.

- No Data Explorer, expanda Data Sets, então expanda Usuário Ceará. As colunas especificadas na consulta aparecem abaixo de Usuário Ceará.

- Arraste "nome" para a primeira célula do detalhe. De forma que fique na segunda linha e apareça o label na primeira.

Para mudar a formatação de qualquer componente do relatório apenas selecione e clique no botão abaixo, em Properties - Font ou outra formatação.

Clique em Preview para ver como está.

Também para uma visualização profissional vá em File - View

5 - Ordenando registros

- Abra o report no layout editor e selecione abaixo properties. No editor clique em Table (abaixo e à esquerda da área da tabela).

- Ao mover o mouse por essa região aparece Table, então clique nesse botão Table. Abaixo aparecerão as propriedades da Tabela.

- Clique na Aba Sorting. Clique em Add para adicionar uma expressão. Em Key selecione "nome" e deixe Accendente.

- Execute o Preview para ver o resultado.

Observe que nomes iniciados com maiúsculas aparecem no início da lista. O BIRT ordena dados tipo string usando códigos UCS2. Em conjuntos de caracteres baseados em ASCII, as letras maiúsculas têm valor semelhante às minúsculas. Para ordenar case-insensitivamente usar:

row["nome"].toUpperCase( )

Na expressão para ordenar.

6 - Adicionando Título ao Relatório

Para isso usaremos um controle Text e tags HTML.

Selecione o controle Text na Paleta e arraste para cima da tabela.

Ao aparecer o diálogo digite:

```
<CENTER><B><span style="font-size: larger"> Relatório de Usuários do DNOCS
</B></span><BR>

<FONT size="small">Somente para uso interno</FONT><BR><BR> Relatório Gerado em
<VALUE-OF>new Date( )</VALUE-OF></CENTER><BR><BR>
```

Utilizando o BIRT com PHP ou simplesmente via Web

Tutorial de Instalação e uso do Tomcat e do BIRT

J2SE Download - http://java.sun.com/javase/downloads/index.jsp

Aqui para baixar somente o JDK (baixei o JDK 5.0 Update 9)

https://sdlc2d.sun.com/ECom/EComActionServlet;jsessionid=65966D4F31A5BD1A572B8A9AB6B2BBDB# (Windows)

https://sdlc2d.sun.com/ECom/EComActionServlet;jsessionid=65966D4F31A5BD1A572B8A9AB6B2BBDB# (Linux)

Tomcat Download - http://tomcat.apache.org/

Direto na versão 5.5.20 - http://tomcat.apache.org/download-55.cgi#5.5.20

http://mirrors.uol.com.br/pub/apache/tomcat/tomcat-5/v5.5.20/bin/apache-tomcat-5.5.20.tar.gz (Linux)

http://mirrors.uol.com.br/pub/apache/tomcat/tomcat-5/v5.5.20/bin/apache-tomcat-5.5.20.exe (Windows)

Caso sua versão do J2SE seja 1.4 baixar também o JDK 1.4 Compatability Package (duprtiot não precisa):

http://mirrors.uol.com.br/pub/apache/tomcat/tomcat-5/v5.5.20/bin/apache-tomcat-5.5.20-compat.tar.gz (Linux)

http://mirrors.uol.com.br/pub/apache/tomcat/tomcat-5/v5.5.20/bin/apache-tomcat-5.5.20-compat.zip (Windows)

Instalação e Configuração do J2SE

Executar

Mover para o diretório /opt

sudo mv jdk-1_5_0_09-linux-i586.bin

chmod u+s

sudo ./jdk-1_5_0_09-linux-i586.bin

Editar seu script de inicialização para adicionar a variável de ambiente JAVA_HOME:

sudo gedit /etc/bash.bashrc

JAVA_HOME=/opt/jdk1.5.0_09

CLASSPATH=.:$JAVA_HOME/lib/tools.jar

CATALINA_HOME=/home/ribafs/prog/tomcat # antes e após o igual não pode haver espaços

PATH=$JAVA_HOME/bin:$PATH

export JAVA_HOME CLASSPATH PATH CATALINA_HOME

Para surtir efeito restart o ambiente gráfico fazendo um logoff ou Ctrl+Alt+Backspace.

Instalando e configurando o Tomcat

- Descompactar apache-tomcat-5.5.20.tar.gz

- Criar a variável de ambiente para o Tomcat:

- Descompactar o apache-tomcat-5.5.20-compat.tar.gz no raiz do diretório do tomcat /home/ribafs/prog/tomcat

Testando o Tomcat

Apontar o browser para o endereço http://localhost:8080

Administrando o Tomcat

No Windows a instalação via Installer já deixa pronto.

Edite o arquivo /home/ribafs/prog/tomcat/conf/tomcat-users.xml e adicionar última linha abaixo:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager"/>
  <role rolename="admin"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="admin" password="admin" fullName="Tomcat Administrator"
roles="admin,manager"/>
</tomcat-users>
```

Com usuário e senha (admin para ambos)

Abrir no browser em http://127.0.0.1:8080/admin/

Entre com admin e admin

Instalando e usando o Visualizador de Relatórios do BIRT

Download - http://download.eclipse.org/birt/downloads/ (procure por Birt Runtime correspondente à versão do seu BIRT):

http://www.eclipse.org/downloads/download.php?file=/birt/downloads/drops/M-R1-2.2M1-200610171021/birt-runtime-2.2M1.zip

Descompactar e copiar o subdiretório WebViewerExample para o diretório /home/ribafs/prog/tomcat/webapps.

Renomear o WebViewerExample para birt-viewer (opcionalmente)

Abra o link no browser http://localhost:8080/manager/html

Entre com admin e admin.

À esquerda clique em /birt-viewer e veja que abre-se o visualizador web de relatórios do BIRT.

Caso deseje abrir um relatório seu já criado, copie para /home/ribafs/prog/tomcat/webapps/birt-

viewer e chame no browser assim:

http://localhost:8080/birt-viewer/frameset?__report=seurelatorio.rptdesign&parametro=valordoparametro

Instalando o Driver JDBC do seu SGBD

Copie para a pasta:

birt-viewer/WEB-INF/platform/plugins/org.eclipse.birt.report.data.oda.jdbc_2.2.0.v20061009-0630/drivers

postgresql-8.1-407.jdbc3.jar (no meu caso)

Observe que "v20061009-0630" vai depender da sua versão.

Abrindo Relatório de Script PHP

Supondo que deseje abrir o relatório "rpt_produtos.rptdesign" na web. Depois de tudo pronto e de ter copiado o relatório para a pasta birt-viewer, basta chamar assim:

<a href=http://localhost:8080/birt_viewer/frameset?__report=rpt_produtos.rptdesign>Relatório</a>

Se preferir formas mais sofisticadas veja estas sugeridas na documentação do BIRT:

```
<?php
$fname = "./clientes.rptdesign";
// Redirect browser
$dest = "http://localhost:8080/birt-viewer/run?__report=";
$dest .= urlencode( realpath( $fname ) );
header("Location: $dest" );
?>
```

Passando parâmetros

```
<?php
$fname = "./clientes.rptdesign";
// Redirect browser
$dest = "http://localhost:8080/birt-viewer/run?__report=";
$dest .= ";sample=" . urlencode( $paramValue );
$dest .= urlencode( realpath( $fname ) );
header("Location: $dest" );
?>
```

Parâmetros de formulários

Deally, we could use PHP to create a UI form that prompts for the report parameters. The BIRT viewer creates this form in Java using information in the report design. Unfortunately, at present, there is no way to retrieve the raw parameter descriptions from the BIRT viewer using a URL. Instead, there are two alternatives we can use.

First, if we know the parameters ahead of time, we can design a custom form in PHP that prompts the user for them. This works if we have a small number of reports, or if we need to create a specialized parameter page for each report anyway.

Second, we can let the BIRT viewer display the parameter page using the frameset URL.
Generating Reports Dynamically

Finally, PHP provides one additional BIRT integration option: the ability to generate report designs dynamically for a specific task. For example, suppose you have a bug tracking system, and you'd like your user to create their own reports via the web. You can ask the user for the columns to display, then use PHP to create a BIRT report design customized to display those columns. PHP is ideal for this: it allows us to insert scripting directly into HTML. Since a BIRT design is XML, and XML is close enough to HTML for PHP, we can "trick" PHP into generating a BIRT report design instead of an HTML page.

To generate a report design, do the following:

- Create a BIRT report design typical of the kind of report you want to create.
- Create a PHP template file that contains this design. Insert the contents of the design file in place of the HTML you'd usually put into a PHP file.
- Redirect PHP's output from the template file into a report design file.
- Use PHP to generate BIRT XML for the table headings and cells the user wants. Use your report design as a template for what is needed. Consult the ROM spec for details on various elements and properties.
- Within the data set in your template, create an SQL query that fetches the required columns. (Works with other data set types as well.)
- Use the code above to redirect the browser to run that report using the BIRT viewer.

The following PHP code redirects the output of a PHP page, template.inc, into a report design called temp.rptdesign:

```
ob_start( );
require "template.inc";
$page = ob_get_contents( );
ob_end_clean( );
$fw = fopen( "temp.rptdesign", "w" );
fputs( $fw, $page, strlen( $page ) );
fclose( $fw );
```

Opções avançadas

http://localhost:8080/birt-viewer/run?__report=report%5CSalesInvoice.rptdesign&OrderNumber=10010

List of Options

The available viewer options include:

```
Option  Description     Values  Default         frameset        run
__format        The output format       html or pdf     html    N       Y
__isnull        Identifies that a report parameter has a null value
Parameter name  None. Required.         N       Y
__locale        Report locale   Java locale value such as en, en-us or ch-zh.
JVM locale      Y       Y
__report        The path to the report design.          None. Required.
Y       Y
__document      The path to the report document.                None. Required.
Y       N
reportParam     Report parameter.       As specified in the report design.
As specified in the report design.      Y       Y
```

Veja os originais para maiores detalhes:

http://www.eclipse.org/birt/phoenix/deploy/viewerSetup.php

http://www.eclipse.org/birt/phoenix/deploy/usingPHP.php

http://www.eclipse.org/birt/phoenix/deploy/viewerUsage.php

Gerador de Relatório iReport

Site oficial - http://jasperforge.org/sf/projects/ireport

Ótimo tutorial sobre o iReport, inclusive com o acesso via web (com java)

www.furutani.eti.br/tutoriais/Mini-Tutorial_Relatorios_Java_JasperReports_e_iReport.pdf

Tutorial de iReport

http://www.javafree.org/javabb/viewtopic.jbb?t=3154

Starting with JasperReports by Gregory Beumer

http://technology.amis.nl/blog/index.php?p=346

Manual do iReport

http://jasperforge.org/sf/wiki/do/viewPage/projects.ireport/wiki/IReportManualV1.2.1

iReport Documentação

http://jasperforge.org/sf/wiki/do/viewPage/projects.ireport/wiki/HomePage

Tutorial JasperReports - IReports

HTML - http://br.geocities.com/robertofurutani/java/Tutorial_JasperReports/

PDF - http://www.furutani.eti.br/tutoriais/Tutorial_Sub_Relatorio_ArrayList.pdf

Fontes - http://br.geocities.com/robertofurutani/java/Tutorial_JasperReports/Tutorial.zip

Outro tutorial em português

http://www.livramento.yu.com.br/tutoriais/ireport.html

A Tutorial on Generating Reports by iReport

http://www.cise.ufl.edu/~otopsaka/CIS4301/ReportDemo/

Gerando relatórios em PHP com fPDF

Site oficial - http://www.fpdf.org

Download - http://fpdf.org/en/download.php

Exemplos diversos - http://fpdf.org/en/script/index.php

Tutoriais - http://fpdf.org/en/tutorial/index.php

Manual em Pt_BR - http://fpdf.org/en/dl.php?id=90

Assistente em JavaScript para ajudar com o fPDF - http://sourceforge.net/projects/fpdfgenerator

Exemplo de uso com acesso ao SGBD PostgreSQL:

Este exemplo encontra-se em - http://fpdf.org/en/script/script11.php

```
<?php
//Example FPDF script with PostgreSQL
//Ribamar FS - ribafs@dnocs.gov.br

define('FPDF_FONTPATH','font/');
```

```
require('fpdf.php');

$pdf=new FPDF();
$pdf->Open();
$pdf->AddPage();
$pdf->SetTitle('Exemplo de Relatório em PDF via PHP');

//Set font and colors
$pdf->SetFont('Arial','B',16);
$pdf->SetFillColor(255,0,0);
$pdf->SetTextColor(255);
$pdf->SetDrawColor(128,0,0);
$pdf->SetLineWidth(.3);

//Table header
$pdf->Cell(20,10,'SIAPE',1,0,'L',1);
$pdf->Cell(50,10,'Nome',1,1,'L',1);

//Restore font and colors
$pdf->SetFont('Arial','',10);
$pdf->SetFillColor(224,235,255);
$pdf->SetTextColor(0);

//Connection and query
$str_conexao='dbname=contabilidade port=5432 user=postgres password=postgres';
$conexao=pg_connect($str_conexao) or die('A conexão ao banco de dados falhou!');
$consulta=pg_exec($conexao,'select * from conveniologin');
$numregs=pg_numrows($consulta);

//Build table
$fill=0;
$i=0;
while($i<$numregs)
{
    $siape=pg_result($consulta,$i,'siape');
    $nome=pg_result($consulta,$i,'nome');
    $pdf->Cell(20,10,$siape,1,0,'R',$fill);
    $pdf->Cell(50,10,$nome,1,1,'L',$fill);
    $fill=!$fill;
    $i++;
}

//Add a rectangle, a line, a logo and some text
$pdf->Rect(5,5,170,80);
$pdf->Line(5,90,90,90);
$pdf->Image('mouse.jpg',185,5,10,0,'JPG','http://www.dnocs.gov.br');
$pdf->SetFillColor(224,235);
$pdf->SetFont('Arial','B',8);
$pdf->SetXY(5,95);
$pdf->Cell(170,5,'PDF gerado via PHP acessando banco de dados - Por Ribamar
FS',1,1,'L',1,'mailto:ribafs@dnocs.gov.br');

$pdf->Output();
?>
```

Tutorial muito bom - PDF Generation With PHP

http://www.devshed.com/c/a/PHP/PDF-Generation-With-PHP/

Gerando Documentos PDF com PHP (fPDF)

http://www.unisinos.br/seminario/sdsl/_arquivos/Gerando_documentos_PDF_com_PHP.pdf

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Geradores_de_Relat%C3%B3rios"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Integração com AJAX

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Conteúdo

- 1 Palavras Acentuadas com AJAX
- 2 Migrating Web-Based PHP Applications to Ajax
- 3 Ajax e PHP - Carregando dados sem refresh
- 4 Exemplos de AJAX em PHP utilizando XAJAX
- 5 Ajax+Php sites
- 6 Busca PHP com Ajax
- 7 Explain the Difference AND How it works between POST and GET in AJAX
- 8 AJAX - Pesquisa Em Tempo Real
- 9 Ajax com PHP - Do Iniciante ao Profissional
- 10 CEP AJAX
- 11 Frameworks com suporte a AJAX
    - 11.1 Dojo
- 12 Artigos Relacionados a AJAX no site da RevistaPHP
- 13 Engatinhando com AJAX e PHP. Parte 1
- 14 Parte 2 do tutorial AJAX + PHP
- 15 Palestras de AJAX

Palavras Acentuadas com AJAX

Dica de Fabrício Nogueira Magri no PHPBrasil

Artigo de Rafael Dohms (PHPDF) sobre acentos

Migrating Web-Based PHP Applications to Ajax

http://www.onlamp.com/pub/a/php/2007/05/10/migrating-web-based-php-applications-to-ajax.html

Ajax e PHP - Carregando dados sem refresh

http://www.imasters.com.br/artigo/3918/javascript/ajax_e_php_-_carregando_dados_sem_refresh//imprimir/

Exemplos de AJAX em PHP utilizando XAJAX

http://www.japs.etc.br/2006/02/exemplos-de-ajax-em-php-utilizando-xajax/

Ajax+Php sites

Estes websites são escolhidos e avaliados por contribuidores de pt.blinklist.com.

http://pt.blinklist.com/tag/ajax+php/

Busca PHP com Ajax

http://www.conexaomx.com.br/artigos/ajax/busca-php_ajax

Explain the Difference AND How it works between POST and GET in AJAX

http://www.planet-source-code.com/vb/scripts/ShowZip.asp?lngWId=8&lngCodeId=1961&strZipAccessCode=tp%2FB19613132

AJAX - Pesquisa Em Tempo Real

http://www.php5.com.br/?php=_scripts/mostra&id_s=105

http://www.php5.com.br/?php=_scripts/mostra&id_s=163

Ajax com PHP - Do Iniciante ao Profissional

http://www.altabooks.com.br/pdf/ajax_php.pdf

CEP AJAX

http://www.imasters.com.br/forum/lofiversion/index.php/t172741.html

Frameworks com suporte a AJAX

Dojo

Download - http://dojotoolkit.org/

Adding Dojo to Your Pages - http://dojotoolkit.org/book/dojo-book-0-4/part-2-out-box-dojo/getting-started/adding-dojo-your-pages

Artigos Relacionados a AJAX no site da RevistaPHP

http://revistaphp.com.br/artigos_resultado.php?c=2

Engatinhando com AJAX e PHP. Parte 1

http://felipetonello.com/blog/2006/11/14/engatinhando-com-ajax-e-php-parte-1/

Parte 2 do tutorial AJAX + PHP

http://felipetonello.com/blog/2007/01/20/engatinhando-com-ajax-e-php-parte-2/

Palestras de AJAX

Palestras e material de AJAX utilizado por Rafael Dohms do PHPDF

Palestras

Script de Auto-complete

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Integra%C3%A7%C3%A3o_com_AJAX"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Integração com Flash

De Wikibooks

< Aplicativos em PHP | Recursos Extras

13.14 - Integração com Flash

Abordando aspectos básicos do uso de Flash em sites com PHP. Edição e pequenos ajustes de arquivos Flash, assim como a criação de pequenas animações.

Este tutorial teve como base a versão 8 professional do Flash mas deve ser válida (com pequenas alterações) para outras versões.

O Flash é uma ferramenta que pode tornar um site bem mais atraente e interativo.

Para isso deve ser usado por profissionais de design com talento e com muita cautela, pois também pode tornar o site pesado para carregar.

Usado com bom senso é um grande aliado, pois traz recursos muito atraentes e difíceis de conseguir com outra ferramenta.

Este pequeno guia não é um tutorial do Flash, mas apenas um guia de uso do Flash para edição de filmes (fontes .FLA).

- Caso o arquivo que vá abrir seja de versão anterior, inicie fazendo uma cópia do mesmo para preservar o original, pois normalmente ao ser editado e salvo em versão mais rcente não mais poderá ser aberto na versão mais antiga.

- Abra o arquvo no Flash e tecle Ctrl+Enter para executar no visualizador. Depois feche para voltar ao Flash.

- Ao selecionar um objeto (movie clip, button ou graphic) podemos alterar suas propriedades e seu ActionScript. Para exibir a janela de propriedades tecle Ctrl+F3 e para exibir a do ActionScript para alterar ou adicionar algun código tecle F9.

- Temos duas ferramentas de seleção (seta preta - Selection Tool - tecla de atalho V e seta branca - Subselection Tool - tecla de atalho A). A selection com um duplo clique seleciona todas as camadas ou componentes de um objeto e pode alterar as propriedades de todos de uma vez, já a Subselection não tem esta função, por exemplo, selecione um segmento de reta e ao clica em um dos extremos pode apenas mover um extremo e deixar o outro fixo. Já a selection moverá o segmento de reta por inteiro.

- Observe que por default a Timeline mostra Scene 1. Ao editar um button (com um duplo clique), ao lado de Scene aparece Symbol 2, pois entramos no modo de edição de Símbolo. Logo abaixo 4 fases do botão, Up, Over, Down e Hit. Up é quando o mouse ao clicar no botão inicia o clique, o down é quando o mouse ao clicar termina o clique e solta o botão. o Over é quando o mouse é

movido passando sobre o botão e o Hit é a porção de área clicável do botão. Para mover de uma região desta para outra arraste o mouse para a outra.

- Algumas vezes certos textos e outros objetos não aparecem no Flash (apenas no visualizador) por mais que selecionemos todos os objetos. Nestes casos tecle Ctrl+L para exibir a biblioteca (Library) com todos os objetos do arquivo. Com um clique sobre o nome ou o ícone selecionamos. Com um duplo clique sobre o ícone selecionamos para edição no centro.

- Ao mostrar o editor de símbolos com um botão teclando Enter ele mostra o comportamento e as demais fases.

- Adicionando ação a um botão

- Primeiro convertemos o objeto em Symbol - Button.

- Depois Selecionamos o objeto e teclamos F9

- Então clicamos no sinal de + da barra de ActionScripts

- Selecionamos Global Functions - Movie Clip Control - on e duplo clique em press

Ficará assim:

```
on (press) {

}
```

- Insira uma linha em branci acima do fecha chaves e Novamente clicar no sinal de +

```
on (press) {
}
```

- Global Functions - Browser/Network - getURL e então digite a URL:

Ficará assim:

```
on (press) {
        getURL("http://ribafs.net", "_blank")

}
```

Como também podemos abrir um arquivo local - getURL ("3Produtividade/index.php","_blank");

- Um recurso importante é fazer o Flash gerar a página HTML que abre um arquivo .SWF.

Vá em File - Publish Settings. Na aba Formats, Type marque HTML.

Agora para publicar (gerar o HTML) basta teclar Shift+F12 ou File - Publish.

- Ctrl+3 exibe a área central de edição de cenários para caber na tela.

-- Integração com Flash e Banco de Dados

- Uma Abordagem muito simples seria criar um codigo em php no qual retornaria na tela com o echo ...

por exemplo:

```
echo "&var1=dadosdavariavel 1&";
echo "&var2=dadosdavariavel 2&";
```

e assim por diante para com demais variaveis ... vc poderia fazer uma chamada no banco de dados e utilizar esta estrutura de variaveis para importarmos para o Flash, salve como dados.php.

para passarmos estas variaveis para o Flash vamos utilizar o LoadVars

Abra o Flash crie dois textos dinâmicos e instâncie eles como os repectivos nomes: Edit1doFlash e Edit2doFlash ... clique no primeiro frame e coloque o seguinte código:

```
//define variavel para carregamento dos dados
carrega = new LoadVars();
//função apos carregamento dos dados...
carrega.onLoad = function(ok) {
      //testa se a carga obteve sucesso
      if (ok) {
              //pega os resultados que o php gera
              Edit1doFlash.text = carrega.var1;
              Edit2doFlash.text = carrega.var2;
      }
//aqui chamamos o php juntamento como o load ... para dar inicio onLoad
carrega.load("http://www.meusite.com.br/dados.php");
//podemos usar este outro com parametros na chamada do php
carrega.load("http://www.meusite.com.br/dados.php?par1=chama1&par2=chama2");

stop();
```

este foi um exemplo bem simples da integração, trazendo dados do php para o flash.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Integra%C3%A7%C3%A3o_com_Flash"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Integração com Java

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Integração Java

A linguagem Java tem se firmado como uma das mais poderosas no ambiente empresarial, em aplicações que exigem alta confiabilidade e flexibilidade, além de integração em ambientes multi-plataforma. A linguagem de script PHP, por sua vez, oferece facilidades na programação de aplicações para a Web, como por exemplo a integração direta com o Servidor Web Apache, além de ser utilizada largamente na Internet atual com o advento do Ajax.

O uso em conjunto das linguagens PHP e Java já começa a ser explorada, e pode trazer a desenvolvedores de aplicações para a Web uma solução que agrega as qualidades de ambas as linguagens.

Este artigo objetiva apontar as etapas necessárias para a integração das linguagens PHP e Java. Apresenta exemplos de configuração do PHP em conjunto com Java em plataformas Windows e Linux.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Integra%C3%A7%C3%A3o_com_Java"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Bibliotecas de Funções

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Conteúdo

- 1 Biblioteca de Funções
- 2 Biblioteca de Funções – 1ª Parte
- 3 Funções utilizadas no tratamento de strings
- 4 Data por Extenso com PHP
- 5 Calculando Dias Úteis entre duas Datas
- 6 Encontrar subString sabendo apenas o início e o final
- 7 Recebebdo o Nome do Diretório que contém o Script Atual

Biblioteca de Funções

Biblioteca de Funções – 1ª Parte

http://www.revistaphp.com.br/print.php?id=149

Funções utilizadas no tratamento de strings

http://www.revistaphp.com.br/print.php?id=151

Data por Extenso com PHP

http://www.revistaphp.com.br/artigo.php?id=148

Calculando Dias Úteis entre duas Datas

Descontados sábados e domingos.

Encontrado no PHPBrasil - http://phpbrasil.com/scripts/script.php/id/3664

```
<?php

/**
* Calcula a quantidade de dias úteis entre duas datas (sem contar feriados)
* @author Marcos Regis
* @param String $datainicial
* @param String $datafinal=null
*/
function dias_uteis($datainicial,$datafinal=null){
  if (!isset($datainicial)) return false;
  if (!isset($datafinal)) $datafinal=time();

$segundos_datainicial =
strtotime(preg_replace("#(\d{2})/(\d{2})/(\d{4})#","$3/$2/$1",$datainicial));
$segundos_datafinal =
strtotime(preg_replace("#(\d{2})/(\d{2})/(\d{4})#","$3/$2/$1",$datafinal));
$dias = abs(floor(floor(($segundos_datafinal-$segundos_datainicial)/3600)/24
 ) );
$uteis=0;
for($i=1;$i<=$dias;$i++){
$diai = $segundos_datainicial+($i*3600*24);
$w = date('w',$diai);
if ($w==0){
//echo date('d/m/Y',$diai)." é Domingo<br />";
}elseif($w==6){
//echo date('d/m/Y',$diai)." é Sábado<br />";
}else{
//echo date('d/m/Y',$diai)." é dia útil<br />";
$uteis++;
```

```
}
}
return $uteis;
}
?>
ex. de uso
<?php
$data='28/02/2007';
echo "Existem ".dias_uteis($data,"17/03/2007")." dias úteis entre $data e hoje";
?>
```

Encontrar subString sabendo apenas o início e o final

```
// Replace a substring of which you only know the beginning and end
function replaceIndefiniteSubstring ($start_needle, $end_needle, $replacement,
$target) {
    $start_pos = strpos($target, $start_needle);
    $end_pos = strpos($target, $end_needle) + strlen($end_needle);
    return substr_replace($target, $replacement, $start_pos, ($end_pos-
$start_pos));
}
```

Site original - http://snippets.dzone.com/posts/show/4211

Recebebdo o Nome do Diretório que contém o Script Atual

```
function getCurrentDirectory() {
    $path = dirname($_SERVER['PHP_SELF']);
    $position = strrpos($path,'/') + 1;
    return substr($path,$position);
}
```

Site original - http://snippets.dzone.com/posts/show/4210

Aqui mais de 3mil pedaços de código - http://snippets.dzone.com/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Bibliotecas_de_Fun%C3%A7%C3%B5es"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Compilando o PHP

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Transformar qualquer Script em um arquivo Executável (.exe)

http://phpbrasil.com/articles/article.php/id/1492

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Compilando_o_PHP"
Page categories: PHP

Aplicativos em PHP/Recursos Extras/Flex

De Wikibooks

< Aplicativos em PHP | Recursos Extras

Conteúdo

- 1 Flex
- 2 O que é Flex?
- 3 Exemplo de Enquete em Flex + PHP e MySQL - Parte I
- 4 Exemplo de Enquete em Flex + PHP e MySQL - Parte II
- 5 O Flex 3 agora é open-source!
- 6 Flex Builder - Tradução
- 7 Flex 2.0 - Criando seu primeiro aplicativo
- 8 Documentação Flex traduzida - Parte 01
- 9 Documentação Flex traduzida - Parte 02
- 10 Flex quick start guide for PHP developers
- 11 Populando um DataGrid no Flex com PHP e MySQL
- 12 Using the Adobe PHP RIA SDK
- 13 24 projetos em Flex 2.0 com código Fonte
- 14 40 Projetos Flex 2 Open-source
- 15 Simple Flex Tutorial
- 16 Flex RemoteObject and AMFPHP 1.9
- 17 Diversos Tutorials sobre Flex
- 18 Tutoriais sobre PHP e Flex
- 19 Flex & PHP Tutorial - Transmitting data using JSON
- 20 Flex and PHP: Party in the Front, Business in the Back
- 21 Flex for PHP Developers
- 22 Lista de discussão no Google - PHP RIA SDK by Adobe
- 23 Gerador de classes PHP para integração FLEX, MySQL e AMFPHP
- 24 Flex 2.0 com PHP e MySQL

Flex

O que é Flex?

Definição no site do fabricante (http://www.adobe.com/br/products/flex/):

"O Adobe® Flex™ 2 é a solução de desenvolvimento de aplicativos mais completa e potente para se criar e fornecer aplicativos avançados de Internet (RIAs) no ambiente empresarial e na Web. Ele permite que as empresas criem aplicativos multimídia personalizados que melhorem significativamente a experiência do usuário, revolucionando o modo como as pessoas interagem com a Web."

Trecho da mensagem de boas vindas da lista flax-brasil (http://br.groups.yahoo.com/group/flex-brasil):

```
3. Como uso o Flex?

Flex é uma tecnolgia criada pela Macromedia para criação de aplicações ricas na
Internet, usando de maneira fácil e rápida você pode criar UI de aplicações em
instantes.
Para começar a usar o Flex como plataforma de desenvolvimento, você precisa ir
ao site da Macromedia na seção de Downloads
(http://www.macromedia.com/downloads/), baixar a  versão Flex Server trial e o
Flex Builder trial.
Após feito o download só instalar o Flex Server integrado com Jrun e inicar o
uso.

4. O que acontece depois que o Flex Server trial expirar?

Após a versão do Flex Server trial expirar ela automaticamente se torna uma
versão Developer, que é totalmente gratuita para pessoas que estão em processo
de desenvolvimento.
A versão de Developers no campo de desenvolvimento não possue limitações, Existe
apenas 2 limitações existente é a requisição de quem ver seu aplicativo feito
com Flex. O  uso é apenas para 5 IPs diferentes, ou seja apenas 5 pessoas podem
visitar ao mesmo tempo a aplicação.
A segunda limitação para quem usa o Flex Server é na hora de compilar  no
servidor o mxml para swf
O swf tem um dispositivo que funciona só 2 ou 3 dias após criado, mas não afeta
em nada a produção.

5. Como funciona o Flex?

Para esse tópico gostaria que visitasse as seguintes matérias no site do
iMasters.
        http://www.imasters.com.br/artigo.php?cn=1931&cc=42
6. Onde acho artigos para minha primeira experiencia com Flex?
        http://www.imasters.com.br/artigo.php?cn=1931&cc=42

7. Quem pode participar da lista?

Quem desejar aprender sobre a tecnologia, as únicas exigências são que para
participar antes de tudo tenha calma na hora de obter a resposta, quem está
praticando em casa mesmo o uso do Flex e quem tem vontade de aprender.

8. Onde encontro documentação para o Flex?
```

```
A documentação você pode encontrar junto com o Flex Server ao instalar.
Mas caso queira uma informação onde sempre está atualizada, peço atenção que
visite o LiveDocs do Flex, sempre é atualizada.

http://livedocs.macromedia.com/flex/
```

No site - http://www.adobe.com/br/products/flex/ encontramos tutoriais, exemplos e ferramentas, como o Flex Builder.

Exemplo de Enquete em Flex + PHP e MySQL - Parte I

http://www.mxstudio.com.br/views.tutorial.php?act=view&cid=12&aid=1104

Exemplo de Enquete em Flex + PHP e MySQL - Parte II

http://www.mxstudio.com.br/views.tutorial.php?act=view&cid=12&aid=1110

O Flex 3 agora é open-source!

http://www.imasters.com.br/artigo/6066/flex/o_flex_3_agora_e_open-source/

Flex Builder - Tradução

Ótimo blog, com muita informação sobre Flex.

http://flex-builder.blogspot.com/search/label/Flex%20%2F%20Flex%20Builder

Flex 2.0 - Criando seu primeiro aplicativo

http://www.imasters.com.br/artigo/4490/flex/flex_20_-_criando_seu_primeiro_aplicativo//imprimir/

Documentação Flex traduzida - Parte 01

http://www.imasters.com.br/artigo/2184/flex/documentacao_flex_traduzida_-_parte_01/

Documentação Flex traduzida - Parte 02

http://www.imasters.com.br/artigo/2361/flex/documentacao_flex_traduzida_-_parte_02//imprimir/

Flex quick start guide for PHP developers

http://www.adobe.com/devnet/flex/articles/php_getstarted.html

Código fonte do exemplo - http://download.macromedia.com/pub/developer/phpguide_export.rar

Populando um DataGrid no Flex com PHP e MySQL

http://www.imasters.com.br/artigo/4685/flex/populando_um_datagrid_no_flex_com_php_e_mysql//imprimir/

Using the Adobe PHP RIA SDK

http://www.adobe.com/devnet/flex/articles/flex_php_sdk.html

Arquivo de exemplo - http://download.macromedia.com/pub/developer/php_ria_sdk_by_adobe_070108.zip

24 projetos em Flex 2.0 com código Fonte

http://www.igorcosta.com/blog/?p=108

40 Projetos Flex 2 Open-source

http://www.igorcosta.com/blog/?p=276

Simple Flex Tutorial

http://www.petefreitag.com/item/490.cfm

Flex RemoteObject and AMFPHP 1.9

O AMFPHP(criado por Patrick Mineault) é uma implemantação em PHP para suportar o protocolo AMF(Action Message Format) facilitando a comunicação entre um SWF(Flash/Flex) e um servidor PHP. Os dados trafegam em formato binário ganhando em performace e segurança. O AMFPHP 1.9 ja suporta o AMF3.

http://sephiroth.it/tutorials/flashPHP/flex_remoteobject/

Diversos Tutorials sobre Flex

http://www.leonardofranca.com.br/index.php/2007/01/16/tutorial-flex-2-e-amfphp-19/pt/

Tutoriais sobre PHP e Flex

http://www.adobe.com/devnet/flex/flex_php.html

Flex & PHP Tutorial - Transmitting data using JSON

http://blog.paranoidferret.com/?p=23

Flex and PHP: Party in the Front, Business in the Back

Desenvolvendo em PHP com Flex via Zend Framework

http://devzone.zend.com/node/view/id/2019

Código fonte do exemplo do tutorial - http://devzone.zend.com/content/mashuptimes.zip

Flex for PHP Developers

http://www.flex.org/php/

Lista de discussão no Google - PHP RIA SDK by Adobe

http://groups.google.com/group/adobe_php_sdk

Gerador de classes PHP para integração FLEX, MySQL e AMFPHP

Clicar na imagem para fazer o download - http://www.flexpr.com.br/gerador/

Flex 2.0 com PHP e MySQL

Apresentação e exemplo.

http://www.flexbrasil.org/palestras/facinter/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Recursos_Extras/Flex"
Page categories: PHP

Aplicativos em PHP/Orientação a Objetos/Teoria e exercícios

De Wikibooks

< Aplicativos em PHP | Orientação a Objetos

Conteúdo

- 1 Tradicional classe inicial - olaMundo
- 2 Construtor
- 3 Variáveis e Métodos Static, onde podemos usar sem instanciar a classe
- 4 Ou chamando a variável diretamente "Estatica::$varStatic"
- 5 Métodos Final
- 6 Método Construtor e Destrutor
- 7 Constantes da Classe
- 8 Clonando Objetos
- 9 InstanceOf (Testar se classe é instância de outra)
- 10 Classes Abstratas
- 11 Interfaces
- 12 Tratamento de erros lógicos - Exceptions
- 13 Singleton
- 14 Pattern Factory
- 15 Listando os métodos de um Objeto
- 16 Listando as variáveis de uma classe
- 17 Listar variáveis de um objeto
- 18 Testando existência de classe e subclasse
- 19 Devolver nome da classe pai para objeto ou classe
- 20 Checar se método da classe existe
- 21 Programação Orientado a Objeto - Design Partners
- 22 Padrões de Projeto - Singleton

Tradicional classe inicial - olaMundo

```
class OlaMundo {
        function OlaMundo(){
                return "Olá Mundo do PHPOO!";
        }
}

$ola = new OlaMundo();
print $ola->OlaMundo();

// Classe Pessoa
class Pessoa {
        private $nome;
        function setNome($nome){
                $this->nome = $nome;
        }

        function getNome(){
                return $this->nome;
        }
}

$joao = new Pessoa();
$joao->setNome("João Brito");
$pedro = new Pessoa();
$pedro->setNome("Pedro Ribeiro");

print '<b><br><br>Classe Pessoa:<br></b>';
print $joao->getNome();
print '<br>';
print $pedro->getNome();
```

Construtor

```
class Construtor {
   function __construct() {
       print "No construtor da Classe";
   }
}

print '<b><br><br>Classe Construtor:<br></b>';
$obj = new Construtor();
print $obj->Construtor;


<h2>Controle de acessos</h2>

class Acessos{
        public $variavelPublic = "Variável Pública<br>";
        protected $variavelProtected = "Variável Protegida<br>";
        private $variavelPrivate = "Variável Privada<br>";

        public function getPublic(){
```

```
                return $this->variavelPublic;
        }

        public function getProtected(){
                return $this->variavelProtected;
        }

        public function getPrivate(){
                return $this->variavelPrivate;
        }

        public function getMetodoPrivate(){
                return Acessos::getPrivate();
        }

}

$especificacaoAcesso = new Acessos();
echo $especificacaoAcesso->getPublic();
echo $especificacaoAcesso->getMetodoPrivate();
//echo $especificaAcesso->getPrivate(); // Dará um erro fatal
```

Variáveis e Métodos Static, onde podemos usar sem instanciar a classe

```
class Estatica{
        static $varStatic = "Variável Estática<br>";

        static function getStatic(){
                return Estatica::$varStatic;
        }
}
```

Ou chamando a variável diretamente "Estatica::$varStatic"

```
echo Estatica::getStatic();
```

Métodos Final

```
class ClasseFinal{
        final function getFinal(){
                echo "Método Final";
        }
}

$classeFinal = new ClasseFinal();
$classeFinal->getFinal();
```

Método Construtor e Destrutor

```
class ContrutorDestrutor{

        private $varMethod;
        function __construct(){
                $this->varMethod = "Construtor()";
                echo "Método {$this->varMethod}<br>";
        }

        function __destruct(){
                $this->varMethod = "Destrutor()";
                echo "Método {$this->varMethod}<br>";
        }
}

$contrutorDestrutor = new ContrutorDestrutor();
unset($contrutorDestrutor);

class ContrutorDestrutorFilho extends ContrutorDestrutor{
        function __construct(){
                parent::__construct();
                echo "Método Filho Construtor<br>";
        }

        function __destruct(){
                parent::__destruct();
                echo "Método Filho Destrutor<br>";
        }
}

echo "<br>";
$contrutorDestrutorFilho = new ContrutorDestrutorFilho();
```

Constantes da Classe

```
class Constante{
        const constante = "Minha Constante";
}
echo Constante::constante;
```

Clonando Objetos

```
class ClasseClonando{
    public $varClone;
    function __construct(){
        $this->varClone = "<br>Php5<br>";
    }

    function __clone(){
        $this->varClone = "Php5 Clone<br>";
    }
}

$classeClonando = new ClasseClonando();
```

```
$cloneClasseClonando = clone $classeClonando;
echo $classeClonando->varClone . "<br>" . $cloneClasseClonando->varClone;
```

InstanceOf (Testar se classe é instância de outra)

```
class TesteInstanceOf
{
//....
}

class ClasseInstanceOf{
    function __construct($obj){
        if ($obj instanceof TesteInstanceOf) {
            echo "Objeto da classe(TesteInstanceOf)<br>";
        } else {
            echo "Não é um objeto da classe(TesteInstanceOf)<br>";
        }
    }
}

$testeInstanceOf = new TesteInstanceOf();
$classeInstanceOf = new ClasseInstanceOf($testeInstanceOf);
```

Classes Abstratas

```
abstract class Abstrata{
    public abstract function setNome($nome);
    public abstract function getNome();
}

class ClasseAbstrata extends Abstrata{
    private $nome;
    public function setNome($newNome){
        $this->nome = $newNome;
    }

    public function getNome(){
        return $this->nome;
    }
}

$classeAbstrata = new ClasseAbstrata();
$classeAbstrata->setNome("Php5");
echo $classeAbstrata->getNome()."<br>";
```

Interfaces

```
interface IPessoa{
    public function setNome($nome);
    public function getNome();
}

interface IPessoaFisica{
    public function setCpf($cpf);
    public function getCpf();
}

interface IPessoaJuridica{
    public function setCnpj($cnpj);
    public function getCnpj();
}

class ClassePessoa implements IPessoa, IPessoaFisica, IPessoaJuridica{
    function __construct($nome, $cpf, $cnpj){
        ClassePessoa::setNome($nome);
        ClassePessoa::setCpf($cpf);
        ClassePessoa::setCnpj($cnpj);
    }

    /* Métodos Set */
    public function setNome($nome){
        $this->nome = $nome;
    }
    public function setCpf($cpf){
        $this->cpf = $cpf;
    }
    public function setCnpj($cnpj){
        $this->cnpj = $cnpj;
    }
    /* Métodos Get */
    public function getNome(){
        return $this->nome;
    }
    public function getCpf(){
        return $this->cpf;
    }
    public function getCnpj(){
        return $this->cnpj;
    }
    function __destruct(){
        echo
ClassePessoa::getNome()."<br>".ClassePessoa::getCpf()."<br>".ClassePessoa::getCn
pj();
    }
}

$classePessoa = new ClassePessoa("Rodrigo", "324.541.588-98", "6545.2101/0001");
```

Tratamento de erros lógicos - Exceptions

```
class BusinessException extends Exception{
    function __construct($msg){
        // Vai para a função construtora do Exception.
        parent::__construct($msg);
    }
}

class Excecao{
    function __construct($nome){
        try {
            if ($nome == "") {
                throw new BusinessException("Nome não pode ser em branco");
            } elseif(strlen($nome) < 5) {
                throw new BusinessException("Nome precisa ter no mínimo 5
letras");
            } elseif(strtolower($nome) == "corinthians") {
                throw new BusinessException("Corinthians campeão");
            } else {
                throw new BusinessException("Paramêtro inválido");
            }
        } catch (BusinessException $businessException) {
            echo $businessException->getMessage();
        }
    }
}

$excecao = new Excecao("Corinthians");
```

Singleton

```
class Singleton {
    private static $instance = null;

    public static function getInstance(){
        if (Singleton::$instance == null) {
            Singleton::$instance = new Singleton();
        }
        return Singleton::$instance;
    }
}

$objA = Singleton::getInstance();
$objB = Singleton::getInstance();
if ($objA == $objB) {
    echo "<br>Instância única";
} else {
    echo "<br>Instâncias diferentes";
}
```

Pattern Factory

```
abstract class AbstractFactory
{
    private $nome;
    private $rendaMensal;

    function __construct($nome, $rendaMensal){
        $this->setNome($nome);
        $this->setRendaMensal($rendaMensal);
    }

    public function setNome($newNome){
        $this->nome = $newNome;
    }

    public function setRendaMensal($newRendaMensal){
        $this->rendaMensal = $newRendaMensal;
    }

    public function getNome(){
        return $this->nome;
    }

    public function getRendaMensal(){
        return $this->rendaMensal;
    }

    public abstract function analisarCredito(); // Boolean
    public abstract function getCategoria(); // String
}

class ClientePadrao extends AbstractFactory{
    function __construct($nome, $rendaMensal){
        parent::__construct($nome, $rendaMensal);
    }

    // Foi declarada no AbstractFactory
    public function analisarCredito(){
        return true;
    }

    // Foi declarada no AbstractFactory
    public function getCategoria(){
        return "Cliente Padrão";
    }
}

class ClienteRisco extends AbstractFactory{
    function __construct($nome, $rendaMensal){
        parent::__construct($nome, $rendaMensal);
    }

    // Foi declarada no AbstractFactory
    public function analisarCredito(){
        return false;
    }
```

```
    // Foi declarada no AbstractFactory
    public function getCategoria(){
        return "Cliente Risco";
    }
}

class ClienteSeguro extends AbstractFactory{
    function __construct($nome, $rendaMensal){
        parent::__construct($nome, $rendaMensal);
    }

    // Foi declarada no AbstractFactory
    public function analisarCredito(){
        return true;
    }

    // Foi declarada no AbstractFactory
    public function getCategoria(){
        return "Cliente com alta credibilidade";
    }
}

class SingletonFactory{
    private static $rendaMedia = 500;
    private static $rendaBaixa = 240;
    private static $instance = null;

    public static function getCliente($nome, $rendaMensal){
        if ($rendaMensal <= SingletonFactory::$rendaBaixa) {
            SingletonFactory::$instance = new ClienteRisco($nome, $rendaMensal);
        } elseif ($rendaMensal > SingletonFactory::$rendaBaixa and
                                $rendaMensal <= SingletonFactory::$rendaMedia) {
            SingletonFactory::$instance = new ClientePadrao($nome,
$rendaMensal);
        } else {
            SingletonFactory::$instance = new ClienteSeguro($nome,
$rendaMensal);
        }

        $clienteAprovacao = "reprovado";
        if (SingletonFactory::$instance->analisarCredito()) {
            $clienteAprovacao = "aprovado";
        }

        echo "<br>Cliente = ".SingletonFactory::$instance->getNome()."<br>";
        echo "Categoria = ".SingletonFactory::$instance->getCategoria()."<br>";
        echo "Crédito = ".$clienteAprovacao;
        echo "<hr>";
    }
}

SingletonFactory::getCliente("Rodrigo", 1977);
SingletonFactory::getCliente("Corinthians", 350);
SingletonFactory::getCliente("John", 220);
```

Listando os métodos de um Objeto

```
class OlaMundo2 {
   // constructor
   function OlaMundo2(){
       return(true);
   }
   // method 1
   function funcao1(){
       return(true);
   }

   // method 2
   function funcao2(){
       return(true);
   }
}

$meus_objetos = new OlaMundo2();
$metodos_classe = get_class_methods(get_class($meus_objetos));

foreach ($metodos_classe as $nome_metodo) {
   echo "$nome_metodo<br>";
}
```

Listando as variáveis de uma classe

```
class Variaveis {

   var $variavel1; // esta não tem valor default...
   var $variavel2 = "xyz";
   var $variavel3 = 100;

   // construtor
   function Variaveis() {
       // mudar algumas propriedades
         $this->var1 = "foo";
         $this->var2 = "bar";
         return true;
   }

}

$minha_classe = new Variaveis();
$variaveis = get_class_vars(get_class($minha_classe));

foreach ($variaveis as $nome => $value) {
   echo "$nome = $value<br>";
}
```

Listar variáveis de um objeto

```
class Point2D {
   var $x, $y;
   var $label;

   function Point2D($x, $y){
       $this->x = $x;
       $this->y = $y;
   }

   function setLabel($label){
       $this->label = $label;
   }

   function getPoint(){
       return array("x" => $this->x,
                    "y" => $this->y,
                    "label" => $this->label);
   }
}

// "$label" is declared but not defined
$p1 = new Point2D(1.233, 3.445);
print_r(get_object_vars($p1));

$p1->setLabel("point #1");
print_r(get_object_vars($p1));
```

Testando existência de classe e subclasse

```
class Foo { var $myVar; }

class Foo_Bar extends Foo { var $myVar2;}

echo class_exists('Foo')."<br>"; //true
echo class_exists('foo')."<br>"; //true
echo class_exists('Foo_Bar')."<br>"; // true
echo get_parent_class('Foo_Bar')."<br>"; // foo (NOTE: NOT Foo!)
```

Devolver nome da classe pai para objeto ou classe

```
class dad {
   function dad(){
   // implemente alguma lógica
   }
}

class child extends dad {
   function child(){
       echo "Eu sou a classe <b>" , get_parent_class($this) , "'s
</b>filho<br>";
```

```
   }
}

class child2 extends dad {
   function child2(){
       echo "Eu também sou a classe <b>" , get_parent_class('child2') , "'s
</b>filho<br>";
   }
}

$foo = new child();
$bar = new child2();
```

Checar se método da classe existe

```
class Foo1 {
  public function bar() {
   echo "Eu sou private Foo1::bar()<br>";
  }
}

class Foo2 {
  private function bar() {
   echo "Eu sou public Foo2::bar()<br>";
  }
}

$f1=new Foo1;
$f2=new Foo2;

if(is_callable(array($f1,"bar"))) {
   echo "Foo1::bar() é acessível<br>";
} else {
   echo "Foo1::bar() não é acessível<br>";
}
if(is_callable(array($f2,"bar"))) {
   echo "Foo2::bar() é acessível<br>";
} else {
   echo "Foo2::bar() não é acessível<br>";
}
if(in_array("bar",get_class_methods($f1))) {
   echo "Foo1::bar() é acessível<br>";
} else {
   echo "Foo1::bar() não é acessível<br>";
}
if(in_array("bar",get_class_methods($f2))) {
   echo "Foo2::bar() é acessível<br>";
} else {
   echo "Foo2::bar() não é acessível<br>";
}
```

Programação Orientado a Objeto - Design Partners

http://www.revistaphp.com.br/print.php?id=140

Padrões de Projeto - Singleton

http://www.revistaphp.com.br/print.php?id=142

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Orienta%C3%A7%C3%A3o_a_Objetos/Teoria_e_exerc%C3%ADcios"
Page categories: PHP

Aplicativos em PHP/Orientação a Objetos/Aplicativos de exemplo

De Wikibooks

< Aplicativos em PHP | Orientação a Objetos

Exemplos de Aplicativos em PHP Orientados a Objetos

Iniciante PHPOO - http://www.ribafs.net/down/appsexemplo/my/iniciantePHPOO.zip

Building Object-Oriented Web Pages with Inheritance in PHP 5

http://www.devshed.com/c/a/PHP/Building-Object-Oriented-Web-Pages-with-Inheritance-in-PHP-5/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Orienta%C3%A7%C3%A3o_a_Objetos/Aplicativos_de_exemplo"
Page categories: PHP

Aplicativos em PHP/Orientação a Objetos/Ferramentas

De Wikibooks

< Aplicativos em PHP | Orientação a Objetos

Conteúdo

- 1 Ferramentas para Trabalhar com PHP Orientado a Objetos
- 2 Classes Selecionadas
  - 2.1 Tiny P2P
  - 2.2 Class: MySqueaks

Ferramentas para Trabalhar com PHP Orientado a Objetos

O Umbrello é uma boa ferramenta para trabalhar com UML com suporte ao PHP - http://uml.sourceforge.net/index.php

Bons rcursos de UML e também gera o código PHP. Até o momento somente com versão for Linux.

O ArgoUML é uma ferramenta que tem suporte a Linux e a Windows que trabalha com UML e tem suporte ao PHP - http://argouml.tigris.org/

Classes Selecionadas

Tiny P2P

Veja só o que esta classe faz:

- Gerenciar informações dos membros da rede - Receber a lista de todos os membros - Receber a lista de todos os amigos de um membro - Contar o número de amigos de um membro - Procurar por membros pelo nome, cidade, país, escola e idade - Adicionar amigos para um dado membro

Acesse o site ótimo repositório de classes PHP:

http://www.phpclasses.org/browse/package/3962.html

Faça o login e baixe a classe. Caso não seja registrado, faça seu registro (rápido e grátis).

Class: MySqueaks

This class can be used to manipulate MySQL database tables and their records data.

It can be used to build SQL queries to perform the basic SELECT, INSERT, UPDATE and DELETE statements from lists of parameters.

The class can wrap also more complex table management queries like: CREATE TABLE, CREATE VIEW, ALTER TABLE, RENAME TABLE, DROP TABLE, and OPTIMIZE.

Faça o login ou registre-se no site abaixo para ter acesso:

http://www.phpclasses.org/browse/package/4028.html

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Orienta%C3%A7%C3%A3o_a_Objetos/Ferramentas"
Page categories: PHP

Aplicativos em PHP/Aplicativos de Exemplo com/MySQL

De Wikibooks

< Aplicativos em PHP | Aplicativos de Exemplo com

15.1 - Aplicativos de Exemplo com MySQL

Vários Aplicativos de Exemplo em PHP com MySQL

http://www.ribafs.net/joomla/index.php?option=com_content&task=category&sectionid=23&id=22&Itemid=63

Sistema de Cadastro com Session

Comentário do autor: "É um sistema de cadastro que utiliza session, tem área administrativa par controle de cadastros do sistema, área par lembrara senha parecido com o do hotmail."

É ótimo e com código bem claro e eficiente.

http://phpbrasil.com/scripts/script.php/id/3658

Formulário para envio de E-mail

http://phpbrasil.com/scripts/script.php/id/3651

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Aplicativos_de_Exemplo_com/MySQL"
Page categories: PHP

Aplicativos em PHP/Aplicativos de Exemplo com/PostgreSQL

De Wikibooks

< Aplicativos em PHP | Aplicativos de Exemplo com

Aplicativos de Exemplo com PostgreSQL

Alguns Aplicativos de Exemplo em PHP com PostgreSQL

http://www.ribafs.net/joomla/index.php?option=com_content&task=category&sectionid=23&id=27&Itemid=63

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Aplicativos_de_Exemplo_com/PostgreSQL"
Page categories: PHP

Aplicativos em PHP/Aplicativos de Exemplo com/SQLite

De Wikibooks

< Aplicativos em PHP | Aplicativos de Exemplo com

15.3 - Aplicativos de Exemplo com SQLite

Um Aplicativo de Exemplo em PHP com SQLite

http://www.ribafs.net/joomla/index.php?option=com_content&task=view&id=61&Itemid=63

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Aplicativos_de_Exemplo_com/SQLite"
Page categories: PHP

Aplicativos em PHP/Apêndices/Análise e Projeto

De Wikibooks

< Aplicativos em PHP | Apêndices

Conteúdo

- 1 Análise de Sistemas na Wikipedia
- 2 Projeto na Wikipedia
- 3 Análise e Projeto
- 4 Algoritmo na Wikipedia
- 5 Lógica e computadores
- 6 Projeto de Bancos de Dados
  - 6.1 Introdução Teórica
  - 6.2 Dicas sobre Campos
  - 6.3 Selecionando o Campo para a Chave Primária
  - 6.4 Chave Primária
  - 6.5 Tipos de Relacionamentos
  - 6.6 Relacionamento Um para Um
  - 6.7 Relacionamento Vários para Vários
  - 6.8 Integridade Referencial
  - 6.9 Normalização de Tabelas
    - 6.9.1 1a Forma Normal
    - 6.9.2 2ª Forma Normal
    - 6.9.3 3ª Forma Normal
  - 6.10 Projeto
  - 6.11 Etapas na Estruturação e Projeto de um Banco de Dados
  - 6.12 Projeto Exemplo
  - 6.13 Hardware indicado para abrigar o servidor de banco de dados
  - 6.14 Software indicado como o servidor dos bancos
  - 6.15 Adotar um modelo e justificá-lo
  - 6.16 Modelo Lógico
  - 6.17 Referências

Análise de Sistemas na Wikipedia

A atividade de análise tem como finalidade realizar estudos de processos afim de encontrar o melhor e mais racional caminho para que a informação possa ser processada. O analista de sistema estuda os diversos sistemas existentes entre hardwares (equipamento), softwares (programas) e o usuário final, seus comportamentos e aplicações, desenvolvendo a partir de então soluções que serão padronizadas e transcritas da forma que o computador possa executar. Gerando o que se chama de softwares (programas), que são executados em hardwares (equipamentos) operados por usuários, individuos, preparados e treinados em procedimentos operacionais padronizados, dotados de conhecimentos do software e hardware para seu trabalho. A partir de então a análise de sistemas é uma profissão, cujas responsabilidades concentram-se na programação e na administração de sistemas computacionais. Cabe a este profissional parte da organização, implantação e manutenção de aplicativos e redes de computadores. Como é uma enfâse, o foco e o núcleo de trabalho está voltado para Administração, levando em conta a area tecnológica em que irá auxiliar.

Fonte: http://pt.wikipedia.org

Projeto na Wikipedia

Segundo o Project Management Institute, projeto é um esforço temporário empreendido para criar um produto ou serviço único. Desta forma, um projeto tem início e fim definidos e resulta em um produto ou serviço de alguma forma diferente de todos os outros anteriormente produzidos.

Seu resultado pode ser:

Um produto ou objeto produzido, quantificável e que pode ser um item final ou um item componente

Uma capacidade de realizar um serviço, como funções de negócios que dão suporte à produção ou à distribuição

Um resultado, como resultados finais ou documentos. Por exemplo, um projeto de pesquisa desenvolve um conhecimento que pode ser usado para determinar se uma tendência está presente ou não ou se um novo processo irá beneficiar a sociedade.

A singularidade é uma característica importante das entregas do projeto. Por exemplo, muitos milhares de prédios de escritórios foram construídos, mas cada prédio em particular é único tem proprietário diferente, projeto diferente, local diferente, construtora diferente, etc. A presença de elementos repetitivos não muda a singularidade fundamental do trabalho do projeto.

Antes do projeto, é comum ainda o trabalhador fazer a preparação de um Anteprojeto, que é o estudo preparatório do projeto.

Nos contextos de software, projeto é usado com o sentido do ato de projetar, de conceber antecipadamente. Neste caso, costuma-se também empregar a palavra design.

Fonte: http://pt.wikipedia.org

Análise e Projeto

1.**Introdução**

2.**Objetivos e Justificativa**

Devem ser definidos o problema existente, o propósito do site (objetivos), por que ele é importante ou necessário (justificativa), para quem ele será útil (usuários), qual o escopo do site (intranet, internet ou extranet) e outras informações que você julgar convenientes e necessárias.

3.**Descrição dos requisitos**

Requisitos de conteúdo - quais informações o site oferece

Requisitos funcionais - quais serviços são oferecidos aos usuários

Requisitos de desenvolvimento - quais recursos materiais e humanos foram utilizados

Requisitos operacionais - quais equipamentos de hardware e quais sistemas de software serão necessário para o web site funcionar. Como será feita a conexão com a internet.

4.**Design - descrição gráfica acompanhada de um texto descrevendo**:

Estrutura organizacional do site - O design da organização envolve a estruturação das diversas páginas que formarão o site, da interligação entre elas.

Aspectos estéticos: Esquema de Layout do banco de dados e dos scripts, estilos de fontes, uso das cores, etc., justificando a sua escolha

5.**Anexos: descrição dos arquivos de HTML, CSS, JavaScript, PHP, etc, que forem utilizados.**

Adaptado do original em - http://www.dimap.ufrn.br/~jair/piws/exercicios2004.html

Faça as perguntas:

- O que iremos desenvolver?
- Como desenvolver?
- Já existe aplicativo similar?
- Se existe, podemos reutilizar algo do mesmo?

Algoritmo e Lógica

Algoritmo na Wikipedia

Um algoritmo é uma sequência finita e não ambígua de instruções para solucionar um problema. Mais especificamente, em matemática, constitui o conjunto de processos (e símbolos que os representam) para efectuar um cálculo. Algoritmos podem ser implementados por programas de

computadores. A palavra algoritmo tem origem no sobrenome, Al-Khwarizmi, do matemático persa do século IX, Mohamed ben Musa, cujas obras foram traduzidas no ocidente cristão no século XII, tendo uma delas recebido o nome "Algorithmi de numero indorum", sobre os algoritmos usando o sistema de numeração decimal (indiano). Outros autores, contudo, defendem a origem da palavra em Al-goreten (raiz - conceito que se pode aplicar aos cálculos).

O conceito de algoritmo é freqüentemente ilustrado pelo exemplo de uma receita, embora muitos algoritmos sejam mais complexos. Eles podem repetir passos (fazer iterações) ou necessitar de decisões (tais como comparações ou lógica) até que a tarefa seja completada. Um algoritmo corretamente executado não irá resolver um problema se o algoritmo estiver incorreto ou não for apropriado ao problema.

Um algoritmo não representa, necessariamente, um programa de computador, e sim os passos necessários para realizar uma tarefa. Sua implementação pode ser feita por um computador, por outro tipo de autômato ou mesmo por um ser humano.

Diferentes algoritmos podem realizar a mesma tarefa usando um conjunto diferenciado de instruções em mais ou menos tempo, espaço ou esforço do que outros. Por exemplo, um algoritmo para se vestir pode especificar que você vista primeiro as meias e os sapatos antes de vestir a calça enquanto outro algoritmo especifica que você deve primeiro vestir a calça e depois as meias e os sapatos. Fica claro que o primeiro algoritmo é mais difícil de executar que o segundo.

Fonte: http://pt.wikipedia.org

Lógica e computadores

A Lógica é extensivamente usada em áreas como Inteligência Artificial, e Ciência da computação.

Nas décadas de 50 e 60, pesquisadores previram que quando o conhecimento humano pudesse ser expresso usando lógica com notação matemática, supunham que seria possível criar uma máquina com a capacidade de pensar, ou seja, inteligência artificial. Isto se mostrou mais difícil que o esperado em função da complexidade do raciocínio humano. programação lógica é uma tentativa de fazer computadores usarem raciocínio lógico e a linguagem de programação Prolog é comumente utilizada para isto.

Na lógica simbólica e lógica matemática, demonstrações feitas por humanos podem ser auxiliadas por computador. Usando demonstração automática de teoremas os computadores podem achar e checar demonstrações, assim como trabalhar com demonstrações muito extensas.

Na ciência da computação, a álgebra booleana é a base do projeto de hardware.

Projeto de Bancos de Dados

O que me levou a compilar este tutorial foi a grande carência de material nesta área, especialmente em português. Se aplicam a qualquer SGBD.

Introdução Teórica

O projeto do banco de dados e também os testes são muito importantes para a eficiência e consistência das informações e do aplicativo. É muito importante gastar algum tempo nesta etapa, pois depois de algum tempo de implantado fica muito trabalhoso alterar estruturas de bancos e aplicativos.

Projetos de banco de dados ineficazes geram consultas que retornam dados inesperados, relatórios que retornam valores sem sentido, etc. Um banco de dados bem projetado fornece um acesso conveniente às informações desejadas e resultados mais rápidos e precisos.

Lembrando:

- SGBDR (Sistema Gerenciador de Bancos de Dados Relacional) – PostgreSQL

- Software de administração de banco de dados para o PostgreSQL – PGAdmin

Informações de bancos de dados relacionais são armazenadas em tabelas ou entidades no Modelo ER (Entidade-Relacionamento).

Dicas sobre Campos

Não armazenar resultado de cálculos ou dados derivados de outros

Armazenar todas as informações (campos) separadamente. Cuidado com campos que contém duas ou mais informações.

Selecionando o Campo para a Chave Primária

A chave primária é o campo ou campos que identificam de forma exclusiva cada registro.

Não são permitidos valores nulos nem duplicados na chave.

Caso a tabela não tenha um campo que a identifique, pode-se usar um campo que numere os registros seqüencialmente.

Dica de Desempenho: O tamanho da chave primária afeta o desempenho das operações, portanto usar o menor tamanho que possa acomodar os dados do campo.

Exemplo

```
Tabela - Clientes
Campo - Nome (atributo)
Chave Primária (Primary Key) - CPF
```

Todos os campos correspondentes a um único CPF juntamente com seus valores formam um Registro ou Linha (Row)

A correta determinação das tabelas, bem como dos campos é algo primordial no sucesso do projeto do banco de dados.

Chave Primária

Obriga que todos os registros terão o campo correspondente à chave primária exclusivo (único - unique). Num cadastro de clientes, todos os clientes cadastrados terão um campo CPF exclusivo. Caso se tente inserir dois clientes com o mesmo CPF o banco não permitirá e emitirá uma mensagem de erro acusando tentativa de violação da chave primária.

Exemplos de Campos indicados para chave primária:

```
CPF
CNPJ
Matrícula de aluno
Matrícula de funcionário
```

Uma chave primária pode ser formada por mais de um campo, quando um único campo não é capaz de caracterizar a tabela.

Cada tabela somente pode conter uma única chave primária.

Relacionamentos – Um banco de dados é formado por várias tabelas. Idealmente essas tabelas devem ser relacionadas entre si para facilitar a troca de informações e garantir a integridade. Para relacionar tabelas usamos chaves existentes nas mesmas.

Tipos de Relacionamentos

```
Um para um
Um para vários
Vários para vários
```

Relacionamento Um para Um

Aquele onde os campos que fazem o relacionamento são chaves primárias. Cada registro de uma tabela se relaciona com apenas um registro da outra tabela.

Este relacionamento não é muito comum.

Exemplo: CorrentistaBanco - Conjuge

Relacionamento Um para Vários

Aquele onde uma tabela tem um campo chave primária (PK) que se relaciona com outra tabela através de um campo chave estrangeira (FK) . É o tipo de relacionamento mais utilizado.

Exemplos:

```
Clientes – Pedidos
Produtos – Itens
Categorias – Itens
Fornecedores – Produtos
NotaFiscal - Produtos
```

Veja que cada registro da esquerda se relaciona com vários registros da direita.

Importante:

O tipo de dados dos campos do relacionamento deve ser igual, assim como o tamanho dos campos e formatos

Chave primária – Chave estrangeira (um – vários)

Relacionamento Vários para Vários

Este tipo de relacionamento não dá para ser implementado no modelo relacional, portanto sempre que nos deparamos com um deles devemos dividir em dois relacionamentos um para vários (criando uma terceira tabela, que armazenará o lado vários dos relacionamentos).

Exemplo:

Pedidos – Produtos (1 --- N e N --- 1)

Cada pedido pode conter vários produtos, assim como cada produto pode estar em vários pedidos. A saída é criar uma tabela que contenha os itens do pedido.

Pedidos – Pedidos_Itens – Produtos

Pedidos 1 - N Pedidos_Itens N - 1 Produtos

Integridade Referencial

Ela garante a integridade dos dados nas tabelas relacionadas. Um bom exemplo é quando o banco impede que se cadastre um pedido para um cliente inexistente, ou impede que se remova um cliente que tem pedidos em seu nome.

Também se pode criar o banco de forma que quando atualizamos ou excluímos o CPF de um cliente ele seja atualizado ou excluído em todos os seus pedidos.

Normalização de Tabelas

Normalizar tabelas e bancos tem o objetivo de tornar o banco mais eficiente, impondo integridade aos dados.

Uma regra muito importante ao criar tabelas é atentar para que cada tabela contenha informações sobre um único assunto, de um único tipo.

1a Forma Normal

Os campos não devem conter grupos de campos que se repetem nos registros.

Exemplo:

Alunos: matricula, nome, data_nasc, serie, pai, mae

Se a escola tem vários filhos de um mesmo casal haverá repetição do nome dos pais. Estão para atender à primeira regra, criamos outra tabela com os nomes dos pais e a matrícula do aluno.

2ª Forma Normal

- Quando a chave primária é composta por mais de um campo. - Devemos observar se todos os campos que não fazem parte da chave dependem de todos os campos que fazem parte da chave.

Caso algum campo dependa somente de parte da chave, então devemos colocar este campo em outra tabela.

Exemplo:

TabelaAlunos

Chave (matricula, codigo_curso)

avaliacao

descricao_curso

Neste caso o campo descricao_curso depende apenas do codigo_curso, ou seja, tendo o código do curso conseguimos sua descrição. Então esta tabela não está na 2ª Forma Normal.

Solução:

Dividir a tabela em duas (alunos e cursos):

TabelaAlunos

Chave (matricula, codigo_curso)

avaliacao

TabelaCursos

codigo_curso descricao_curso

3ª Forma Normal

Quando um campo não é dependente diretamente da chave primária ou de parte dela, mas de outro campo da tabela que não pertence à chave primária. Quando isso ocorre esta tabela não está na terceira forma normal e a solução é dividir a tabela.

Lembrando: Engenharia Reversa (parte de um banco existente ou de um script sql e gera o modelo).

Projeto

Fases do Projeto do Banco de Dados

Modelo Conceitual

Modelo Lógico

Modelo Físico (faz parte apenas da implementação)

Observação.: Trataremos apenas de novos projetos.

Modelo Conceitual – Define apenas quais os dados que aparecerão no banco de dados, sem se importar com a implementação do banco. Para essa fase o que mais se utiliza é o DER (Diagrama Entidade-Relacionamento).

Modelo Lógico – Define quais as tabelas e os campos que formarão as tabelas, como também os campos-chave, mas ainda não se preocupa com detalhes como o tipo de dados dos campos, tamanho, etc.

Etapas na Estruturação e Projeto de um Banco de Dados

Problemas a serem solucionados com o banco de dados

Determinar as tabelas necessárias (cada uma com um único assunto exclusivo)

Determinar os campos de cada tabela

Criar um DER

Verificar a estimativa do crescimento do banco e preparar-se para isso

Investigar como são armazenadas as informações atualmente e recolher a maior quantidade de informações para o projeto

Adotar um modelo e justificá-lo

(Os itens acima fazem parte do Modelo Conceitual e abaixo do Lógico)

Determinar a chave primária de cada tabela. Pode haver tabela sem chave primária.

Determinar os relacionamentos e seus tipos

Obs.: Somente quando da implementação (modelo físico) serão tratados os detalhes internos de armazenamento. O modelo físico é a tradução do modelo lógico para a linguagem do SGBDR adotado.

Projeto Exemplo

Vamos elaborar um exemplo de projeto de banco de dados que será um controle de funcionários.

Controle de Funcionários

Modelo Conceitual

Problemas a serem solucionados com o banco de dados

Atualmente os funcionários são cadastrados em fichas de papel guardadas em pastas. Isso acarreta

dificuldade no resgate de informações e fragilidade das mesmas.

O objetivo do banco de dados será armazenar as informações sobre os funcionários, possibilitando consultas ágeis que retornem as informações de maneira rápida e prática.

Determinar as tabelas necessárias (cada uma com um único assunto exclusivo)

Utilizaremos apenas a tabela funcionarios, contendo todos os dados dos funcionários.

Determinar os campos da tabela

Os campos serão: cpf, nome, email, endereco, cep, cidade, estado, fone, celular, conjuge, filhos.

Criar um DER (uma ótima ferramenta é o plugin para Eclipse Azzurri Clay).

```
funcionarios
cpf
nome
...
```

Hardware indicado para abrigar o servidor de banco de dados

O hardware mais adequado deve ter um disco rígido rápido, boa quantidade de memória RAM, uma boa placa-mãe e um bom processador. As especificações dependem de cada caso.

Software indicado como o servidor dos bancos

Como SGBDR a indicação vai para o PostgreSQL,por ser robusto, estável, bom desempenho, boa documentação, grande comunidade através da Internet e Licença free e open-source para todos os usos.

Como sistema operacional para o servidor a indicação vai para o Linux Debian.

Como sistema operacional para os clientes a indicação vai para o Linux Ubuntu.

Verificar a estimativa do crescimento do banco e preparar-se para isso

A empresa é pequena mas tem perspectiva de crescimento a médio e longo prazo...

Adotar um modelo e justificá-lo

O modelo adotado foi o MER (Modelo Entidade-Relacionamento), por atender às necessidades do cliente e ser o modelo padrão de mercado atualmente.

Modelo Lógico

Teremos apenas uma tabela, funcionarios, cuja chave primária é o campo CPF.

Aplicar as 3 formas normais para testar a coerência das tabelas.

Primeira - Os campos não devem conter grupos de campos que se repetem nos registros.

Cada funcionário terá todos os campos diferentes, exceto alguns campos isolados, como cep, cidade, mas não temos grupos de campos que se repetem.

Sugestão: Ter cidade, cep e estado em tabelas diferentes e relacionadas, deixando em funcionários apenas combos onde seriam selecionados.

Mas tanto o campo conjuge quanto o filhos devem ir para tabelas separadas, já que cônjuge pode armazenar várias informações e filhos também.

Então teremos as tabelas:

funcionarios, conjuges e filhos, de acordo com o DER abaixo.

Utilizando o DBDesigner, o DER acima gerou o script abaixo:

```
CREATE TABLE funcionarios (
  cpf VARCHAR NOT NULL,
  nome VARCHAR NULL,
  endereco VARCHAR NULL,
  cep VARCHAR NULL,
  cidade VARCHAR NULL,
  estado VARCHAR NULL,
  cod_conjuge VARCHAR NULL,
  cod_filhos VARCHAR NULL,
  PRIMARY KEY(cpf)
);

CREATE TABLE func_filhos (
  codigo INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  funcionarios_cpf VARCHAR NOT NULL,
  nome VARCHAR NULL,
  cpf_funcionario VARCHAR NULL,
  data_nascimento DATE NULL,
  PRIMARY KEY(codigo),
  FOREIGN KEY(funcionarios_cpf)
    REFERENCES funcionarios(cpf)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
);

CREATE TABLE func_conjuge (
  cpf_funcionario VARCHAR NOT NULL AUTO_INCREMENT,
  funcionarios_cpf VARCHAR NOT NULL,
  nome INTEGER NULL,
  data_nascimento DATE NULL,
  PRIMARY KEY(cpf_funcionario),
  FOREIGN KEY(funcionarios_cpf)
    REFERENCES funcionarios(cpf)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
);

funcionarios 1 - 1 conjuge
funcionarios 1 - N filhos
```

Tutorial sobre uso do DBDesigner com PostgreSQL: http://wwwpostgresql.org.br

Segunda Forma Normal – Quando a chave primária é composta por mais de um campo.

Não é o caso, pois CPF é suficiente para representar cada funcionário.

Terceira Forma Normal- Quando um campo não é dependente diretamente da chave primária ou de

parte dela, mas de outro campo da tabela que não pertence à chave primária. Não é o caso.

Referências

- O Modelo Relacional de Dados (em cinco artigos, de Júlio Battisti)
http://www.imasters.com.br/artigo.php?cn=2419&cc=149

- Conceitos Fundamentais de Banco de Dados (de Ricardo Rezende)
http://www.sqlmagazine.com.br/Colunistas/RicardoRezende/02_ConceitosBD.asp

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/An%C3%A1lise_e_Projeto"
Page categories: PHP

Aplicativos em PHP/Apêndices/Design

De Wikibooks

< Aplicativos em PHP | Apêndices

Conteúdo

- 1 Web Design, o artista da Web
- 2 Design na Wikipedia
- 3 Introdução ao web design
  - 3.1 Prólogo aos primeiros passos no web design
  - 3.2 1 - http://www.criarweb.com/artigos/816.php
  - 3.3 2 - http://www.criarweb.com/artigos/817.php
  - 3.4 3 - http://www.criarweb.com/artigos/818.php
- 4 Dicas para trabalhar com CSS
- 5 Destruir 2 mitos do web design
- 6 Usabilidade
- 7 7 dicas simples de tipografia para melhorar o visual de seu blog ou de seus documentos
- 8 Adicionando a Busca da Wikipedia no seu Site
- 9 Estamos atendendo ao usuário final?

Web Design, o artista da Web

Design na Wikipedia

Entende-se o design (em alguns casos projeto ou projecto) como o esforço criativo relacionado à configuração, concepção, elaboração e definição de algo, como um objeto, uma imagem, entre outros, em geral voltados a uma determinada função. De uma forma ampla o termo design, porém, refere-se à concepção de uma solução prévia para um problema. Mas em uma acepção mais específica, design se refere à profissão da pessoa que projeta. Como tal, tem diversas especializações, de acordo com o tipo de coisa a projetar. O profissional que trabalha na área de design é chamado, portanto, de designer, visto a palavra pertencer à língua inglesa e normalmente não se traduz (ver o problema etimológico).

Design é, portanto, um esforço criativo através do qual se projetam todo tipo de coisas, incluindo utensílios, vestimentas, peças gráficas, livros, máquinas, ambientes e (recentemente) também interfaces de programas.

As especializações mais comuns são o design de produto, design gráfico e o design de moda. O design está intimamente ligado às artes aplicadas, à arquitetura, e à engenharia, mas a concorrência profissional muitas vezes leva à animosidade entre essas áreas[Carece de fontes?].

O termo deriva, originalmente, de designare, palavra em latim, sendo mais tarde adaptado para o inglês design. Houve uma série de tentativas de tradução do termo, mas os possíveis nomes como projética industrial acabaram em desuso.

Introdução ao web design

Bons tutoriais no ótimo site CriaWeb.

Prólogo aos primeiros passos no web design

http://www.criarweb.com/artigos/816.php

http://www.criarweb.com/introducao_webdesign/

1 - http://www.criarweb.com/artigos/816.php

2 - http://www.criarweb.com/artigos/817.php

3 - http://www.criarweb.com/artigos/818.php

Dicas para trabalhar com CSS

http://webbemfeita.com/css/dicas-para-trabalhar-com-css/

Destruir 2 mitos do web design

http://www.criarweb.com/artigos/804.php

Usabilidade

Uma boa coleção de links sobre usabilidade - http://www.usableweb.com/

7 dicas simples de tipografia para melhorar o visual de seu blog ou de seus documentos

http://www.efetividade.net/2006/10/28/5-dicas-de-tipografia-para-melhorar-o-visual-de-seu-blog-ou-de-seus-documentos/

Adicionando a Busca da Wikipedia no seu Site

O trecho de código abaixo mostrará um pequeno formulário para busca direto na Wikipedia, com opção de vários idiomas.

A imagem pode ser facilmente conseguida no Google.

```
<?xml version="1.0" encoding="UTF-8"?>
<h3>Wikipédia</h3><img src='Wikipedia-logo.jpg' width='150' heigth='200'
border='0'><br>
<form id="searchform" action="http://www.wikipedia.org/search-redirect.php"
name="searchform" target="_blank">
<input type="text" accesskey="f" name="search" id="searchInput" /><br>
<select style="margin: 0pt 0.4em; padding: 0pt; vertical-align: top;"
name="language">
<option lang="de" xml:lang="de" value="de">Deutsch</option>
<option lang="en" xml:lang="en" value="en">English</option>
<option lang="el" xml:lang="el" value="el">Ελληνικά</option>
<option lang="es" xml:lang="es" value="es">Español</option>
<option lang="fr" xml:lang="fr" value="fr">Français</option>
<option lang="it" xml:lang="it" value="it">Italiano</option>
<option lang="nl" xml:lang="nl" value="nl">Nederlands</option>
<option lang="ja" xml:lang="ja" value="ja">日本語</option>
<option lang="pl" xml:lang="pl" value="pl">Polski</option>
<option lang="pt" selected="true" xml:lang="pt" value="pt">Português</option>
<option lang="sv" xml:lang="sv" value="sv">Svenska</option></select>
```

```
<input type="submit" value="Localizar" id="searchGoButton" class="searchButton"
name="go" /></form>
```

Estamos atendendo ao usuário final?

http://www.vivaolinux.com.br/artigos/impressora.php?codigo=6827

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Design"
Page categories: PHP

Aplicativos em PHP/Apêndices/Hospedagem e Domínio

De Wikibooks

< Aplicativos em PHP | Apêndices

1. Domínio
2. Hospedagem de Site com cPanel

Basicamente para publicar um site na Internet precisamos desses dois serviços: uma hospedagem que abrigará o material do site e um serviço que administra o domínio (o nome, o endereço do site).

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Hospedagem_e_Dom%C3%ADnio"
Page categories: PHP

Aplicativos em PHP/Apêndices/Hospedagem e Domínio/Domínio

De Wikibooks

< Aplicativos em PHP | Apêndices | Hospedagem e Domínio

Domínio

O domínio é o que dá o endereço do site, no caso a URL, coisa deste tipo http://www.google.com.br ou http://pt.wikibooks.org.

Temos também os sub-domínios, que são criados em domínios para facilitar a memorização e também para organizar a administração da hospedagem.

Existem servidores de hospedagem (web hosting) free e comerciais (pagos). Caso precise de maior estabilidade e segurança os comerciais são mais indicados (geralmente). Mas se não quer gastar e pretende abrigar um site apenas para testes os gratuitos podem ajudar.

Um subdomínio está relacionado a um diretório do domínio principal. Exemplo:

Domínio - google.com Sub-domínio - video.google.com, que é um diretório do domínio principal, geralmente o diretório video, mas não obrigatoriamente.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Hospedagem_e_Dom%C3%ADnio/Dom%C3%ADnio"
Page categories: PHP

Aplicativos em PHP/Apêndices/Hospedagem e Domínio/Hospedagem de Site com cPanel

De Wikibooks

< Aplicativos em PHP | Apêndices | Hospedagem e Domínio

Conteúdo

- 1 Hospedagem de Site com cPanel
  - 1.1 CPanel Tutorial
- 2 Uso de clientes de FTP
- 3 Sua própria empresa de hospedagem
- 4 Hospedagem de Site com cPanel
  - 4.1 CPanel Tutorial
- 5 Uso de clientes de FTP
- 6 Sua própria empresa de hospedagem
- 7 Ferramentas para Administração de Hospedagens
  - 7.1 WinSCP
  - 7.2 putty
  - 7.3 Krusader
  - 7.4 SSH
  - 7.5 Fantastico
  - 7.6 MySQL Quick Admin
- 8 Hospedagem Grátis com Suporte a PHP e MySQL

Hospedagem de Site com cPanel

Hospedagem e Domínio

Para ter um site publicado na Internet de forma que qualquer pessoa possa visitá-lo, você precisa ter um domínio (nome do site, por exemplo www.meusite.org.br). Para isso deverá pagar uma anuidade por esse domínio, ou ainda poderá ter um sub-domínio em um servidor de hospedagem, ou mesmo em um serviço gratuito. Também precisará de um abrigo para o seu site, os arquivos. Este abrigo chama-se servidor de hospedagem, web hosting.

Existem diversos servidores de hospedagem free:

- A exemplo do Geocities-Yahoo (http://geocities.yahoo.com/), que não tem suporte a PHP e MySQL

- E outros que oferecem suporte a PHP e MySQL mas não suportam Joomla, como é o caso do

phpnet.us (http://www.phpnet.us/), e também do

1500mb.com (http://www.1500mb.com/), que dá 1.5 GB de espaço

- Temos também alguns rartos com suporte a PHP, MySQL e ao Joomla/Mambo, como o

www.byethost.com. (250 MB de espaço) e o

www.funpic.org (2,5GB de espaço com banda e também oferece e-mail).

Em ambos devemos descompactar o joomla localmente e enviar por FTP descompactado.

Servidores com CPanel geralmente têm mais recursos, um bom deles é que você envia o arquivo compactado e descompacta lá mesmo no servidor, através do gerenciador de arquivos.

Obs.: vale lembrar que estes sites mudam muito e nada garante que quando você for acessar um dos acima ele ainda esteja por lá.

Caso esteja querendo um site estável, com bom suporte, talvez deva evitar hospedagem gratuita. Procure por alguém que já utilize um bom serviço de hospedagem, uma lista, um forum, etc. Se possível um serviço na sua cidade, para economizar nas ligações para o suporte. Mas o mais importante é a seriedade do serviço.

CPanel Tutorial

O CPanel é um painel para administração de sites em servidores de hospedagem. É o mais popular programa de administração de servidores de hospedagem (para os clientes), especialmente voltado para servidores Linux.

Com o endereço fornecido pelo servidor de hospedagem acesse até que aparece a janela de login:

Quando aparece a tela principal:

Veja que oferece diversos recursos.

- Administração de Arquivos e Pastas:

- criação e edição

- renomear
- permissões
- upload

Aqui para abrir uma pasta clique no ícone da mesma. Para abrir o menu de configurações para uma pasta ou arquivo clique no nome. Exemplo: para editar o arquivo index.php, clique no nome do mesmo e em Edite File à direita. Também podemos fazer o upload de arquivos para o servidor, clicando em Upload file(s) Podemos voltar para pastas de nível acima clicando em Up one level. Podemos criar novas pastas e arquivos também.

Administração dos SGBDs MySQL e PostgreSQL: criar bancos criar usuários e senhas associar usuários a bancos administrar bancos com phppgadmin ou phpmyadmin

Para criar um banco no MySQL clique em Banco de Dados MySQL para abrir:

MySQL Account Maintenance

Current Databases:

ribafsn_joomla Delete Check Repair

Users in joomla

ribafsn_joomla (Privileges: ALL PRIVILEGES) Delete

Connection Strings

Perl
```
$dbh =
DBI->connect("DBI:mysql:ribafsn_joomla:localhost","ribafsn_joomla","<PASSWORD
HERE>");
```

PHP
```
$dbh=mysql_connect ("localhost", "ribafsn_joomla", "<PASSWORD HERE>") or die ('I
cannot connect to the database because: ' . mysql_error());
mysql_select_db ("ribafsn_joomla");
```

ribafsn_wordpress Delete Check Repair

Users in wordpress

ribafsn_wordpres (Privileges: ALL PRIVILEGES) Delete

Connection Strings

Perl
```
$dbh =
DBI->connect("DBI:mysql:ribafsn_wordpress:localhost","ribafsn_wordpres","<PASSWORD
HERE>");
```

PHP
```
$dbh=mysql_connect ("localhost", "ribafsn_wordpres", "<PASSWORD HERE>") or die ('I
cannot connect to the database because: ' . mysql_error());
mysql_select_db ("ribafsn_wordpress");
```

New Database: [ ] Create Database

1 - Digitar o nome do banco em New Database (ex.: joomla) e clicar em Create Database

2 - Digite o nome do usuário em Username (ex.: joomla) e a senha em Password (ex.: joomla) e clique em Create User.

Em cada operação há que se voltar para continuar.

3 - Após criar um banco e usuário e senha temos que Adicionar o usuário ao banco criado. Para isso selecionamos o usuário à esquerda e o banco à direita. Veja que ao nome que você digitou é adicionado automaticamente um prefixo pelo servidor e também ao nome do banco. Este prefixo é o nome do usuário no servidor.

Administração de e-mails:

criação e edição

redirecionamento

Administração do domínio e sub-domínios:

criação e edição

redirecionamento (apontamento)

Administração de FTP:

criação e edição de contas

Estatísticas de uso:

espaço em disco

banda

e-mails

sub-domínios

Informações sobre o Servidor:

status dos serviços

versões dos serviços

caminho do perl e do sendmail

- Também podemos Fazer Backup do site e Restaurar opções para backup completo, da pasta home ou dos bancos de dados do MySQL.

- Acesso via SSH

Um recurso muito útil para quem mexe com Linux. Com este podemos alterar o dono dos arquivos de um diretório, coisa que não se pode fazer pelo CPanel nem por um cliente de FTP.

Acesso Telnet shell

For security reasons, shell access is not enabled by default. In order to activate shell access on your account, you will need to fax or mail a copy of your driver's license, passport or other photo id to customer service.

Sua seção foi logada. Qualquer tentativa de copiar/excluir/ver arquivos que não são seus resultará na exclusão de sua conta!

Press to Start!

jta: ribafs.net 22

File Edit Terminal Help

Java Applet Window

```
.bash_logout                          .dns                public_html/
.bash_profile                         .emacs              tmp/
.bashrc                               etc/                .trash/
.contactemail                         .fantasticodata/    www@
.cpaddons/                            .htpasswds/
ribafsn@ribafs.net [~]# cd public_html
ribafsn@ribafs.net [~/public_html]# ls
./        .htaccess  joomla/        teste.php  _vti_cnf/      _vti_pvt/
../       images/    postinfo.html  tiago/     _vti_inf.html  _vti_txt/
cgi-bin/  index.php  _private/      _vti_bin/  _vti_log/      wordpress/
ribafsn@ribafs.net [~/public_html]# cd joomla
ribafsn@ribafs.net [~/public_html/joomla]# ls
./                     COPYRIGHT.php  index.php     offlinebar.php
../                    editor/        INSTALL.php   offline.php
administrator/         globals.php    language/     pathway.php
cache/                 help/          LICENSE.php   robots.txt
CHANGELOG.php          htaccess.txt   mainbody.php  templates/
components/            images/        mambots/
configuration.php*     includes/      media/
configuration.php-dist index2.php     modules/
ribafsn@ribafs.net [~/public_html/joomla]#
```

Connected to ribafs.net ssh online

Uso de clientes de FTP

Tutorial de Uso do gFTP (Linux)

Pode ser instalado pela opção Adicionar/Remover do Ubuntu ou por outro método em outra distro.

Este é um bom cliente de FTP, com os recursos necessários para se administrar sites remotamente.

Para a conexão digita-se o nome ou IP do do servidor remoto em Máquina, digita-se o nome do usuário em Usuário, digita-se a porta em Porta (apenas se for diferente da padrão, 21) e clica-se no botão de Conexão (dois computadores). Após clicar será solicitada a senha.

Uma boa dica é salvar os dados da conexão nos Marcadores (menu).

Como Alterar as Permissões de Arquivos e Pastas:

Clica-se sobre a pasta do lado direito (remoto) com o botão direito e seleciona-se Chmod...

No caso do Joomla, os diretórios precisam ter permissões como as acima para a instalação.

Tutorial de Uso do SmartFTP (Windows)

O SmartFTP é um bom cliente de FTP para Windows free.

Encontra-se em - http://www.smartftp.com/

Para fazer a conexão execute o programa:

Clique na caixa à direita de Address e selecione o protocolo FTP.

Digite à direita o nome ou IP do servidor, o Login e a Senha. Então clique na seta verde (Go) para conectar. Opcionalmente podemos clicar no último botão da imagem acima (New Remote Browser) para abrir o diálogo e entrar com os mesmos dados.

Então aparecerá a área remota do servidor à direita:

Alterando as Permissões de Arquivos

Após a conexão clique no arquivo ou pasta que deseja ter as permissões alteradas com o botão direito e Properties/CHMOD ou tecle F7 que abre o diálogo:

Veja que podemos tanto digitar o número quanto setar cada uma das permissões. Podemos também selecionar vários e alterar as permissões de todos ao mesmo tempo.

Sua própria empresa de hospedagem

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=6407

Hospedagem de Site com cPanel

Hospedagem e Domínio

Para ter um site publicado na Internet de forma que qualquer pessoa possa visitá-lo, você precisa ter um domínio (nome do site, por exemplo www.meusite.org.br). Para isso deverá pagar uma anuidade por esse domínio, ou ainda poderá ter um sub-domínio em um servidor de hospedagem, ou mesmo em um serviço gratuito. Também precisará de um abrigo para o seu site, os arquivos. Este abrigo chama-se servidor de hospedagem, web hosting.

Existem diversos servidores de hospedagem free:

- A exemplo do Geocities-Yahoo (http://geocities.yahoo.com/), que não tem suporte a PHP e

MySQL

- E outros que oferecem suporte a PHP e MySQL mas não suportam Joomla, como é o caso do

phpnet.us (http://www.phpnet.us/), e também do

1500mb.com (http://www.1500mb.com/), que dá 1.5 GB de espaço

- Temos também alguns rartos com suporte a PHP, MySQL e ao Joomla/Mambo, como o

www.byethost.com. (250 MB de espaço) e o

www.funpic.org (2,5GB de espaço com banda e também oferece e-mail).

Em ambos devemos descompactar o joomla localmente e enviar por FTP descompactado.

Servidores com CPanel geralmente têm mais recursos, um bom deles é que você envia o arquivo compactado e descompacta lá mesmo no servidor, através do gerenciador de arquivos.

Obs.: vale lembrar que estes sites mudam muito e nada garante que quando você for acessar um dos acima ele ainda esteja por lá.

Caso esteja querendo um site estável, com bom suporte, talvez deva evitar hospedagem gratuita. Procure por alguém que já utilize um bom serviço de hospedagem, uma lista, um forum, etc. Se possível um serviço na sua cidade, para economizar nas ligações para o suporte. Mas o mais importante é a seriedade do serviço.

CPanel Tutorial

O CPanel é um painel para administração de sites em servidores de hospedagem. É o mais popular programa de administração de servidores de hospedagem (para os clientes), especialmente voltado para servidores Linux.

Com o endereço fornecido pelo servidor de hospedagem acesse até que aparece a janela de login:

Quando aparece a tela principal:

Banda utilizada (neste mês)	0.00 Megabytes
Contas de Email	1 / unlimited
Redirecionamentos de Email	0
Auto-respostas	0
Mailing Lists	0 / unlimited
Filtros de Email	0
Contas de FTP	0 / unlimited
Informações do Servidor :	
Sistema Operacional	Linux
Stats de Serviços	Clique para ver
Versão do Kernel	2.6.9-42.0.2.ELsmp
Machine Type	i686
Versão do Apache	1.3.36 (Unix)
Verção do PERL	5.8.5
Path do PERL	/usr/bin/perl
Path para o sendmail	/usr/sbin/sendmail
Módulos do Perl	Clique para ver
Versão do PHP	4.4.2
Versão do MySQL	4.1.21-standard
cPanel Build	10.8.2-RELEASE 119
Theme	cPanel X v2.5.0
Documentação	Clique para ver

Redirects | Extensões Frontpage | Web/FTP Stats | Raw Access Logs | Gerenciador do Raw Log

Log de Erros | Stats dos Subdomínios | Chatroom | PhpMyChat | Bulletin Board

Central Cgi | Livraria de Scripts | Agora Shopping Cart | Trabalhos Cron | Ferramentas de Rede

Tipos MIME | Apache Handlers | Gerenciador de chaves OpenPGP | Proteção HotLink | Index Manager

IP Deny Manager | SSL Manager | Statistics Software Configuration | Fantastico

Veja que oferece diversos recursos.

- Administração de Arquivos e Pastas:
- criação e edição
- renomear
- permissões
- upload

Aqui para abrir uma pasta clique no ícone da mesma. Para abrir o menu de configurações para uma pasta ou arquivo clique no nome. Exemplo: para editar o arquivo index.php, clique no nome do mesmo e em Edite File à direita. Também podemos fazer o upload de arquivos para o servidor, clicando em Upload file(s) Podemos voltar para pastas de nível acima clicando em Up one level. Podemos criar novas pastas e arquivos também.

Administração dos SGBDs MySQL e PostgreSQL: criar bancos criar usuários e senhas associar usuários a bancos administrar bancos com phppgadmin ou phpmyadmin

Para criar um banco no MySQL clique em Banco de Dados MySQL para abrir:

MySQL Account Maintenance

Current Databases:

ribafsn_joomla Delete Check Repair

Users in joomla

ribafsn_joomla (Privileges: ALL PRIVILEGES) Delete

Connection Strings

Perl
```
$dbh =
DBI->connect("DBI:mysql:ribafsn_joomla:localhost","ribafsn_joomla","<PASSWORD
HERE>");
```

PHP
```
$dbh=mysql_connect ("localhost", "ribafsn_joomla", "<PASSWORD HERE>") or die ('I
cannot connect to the database because: ' . mysql_error());
mysql_select_db ("ribafsn_joomla");
```

ribafsn_wordpress Delete Check Repair

Users in wordpress

ribafsn_wordpres (Privileges: ALL PRIVILEGES) Delete

Connection Strings

Perl
```
$dbh =
DBI->connect("DBI:mysql:ribafsn_wordpress:localhost","ribafsn_wordpres","<PASSWORD
HERE>");
```

PHP
```
$dbh=mysql_connect ("localhost", "ribafsn_wordpres", "<PASSWORD HERE>") or die ('I
cannot connect to the database because: ' . mysql_error());
mysql_select_db ("ribafsn_wordpress");
```

New Database: [ ] Create Database

1 - Digitar o nome do banco em New Database (ex.: joomla) e clicar em Create Database

2 - Digite o nome do usuário em Username (ex.: joomla) e a senha em Password (ex.: joomla) e clique em Create User.

Em cada operação há que se voltar para continuar.

3 - Após criar um banco e usuário e senha temos que Adicionar o usuário ao banco criado. Para isso selecionamos o usuário à esquerda e o banco à direita. Veja que ao nome que você digitou é adicionado automaticamente um prefixo pelo servidor e também ao nome do banco. Este prefixo é o nome do usuário no servidor.

Administração de e-mails:

criação e edição

redirecionamento

Administração do domínio e sub-domínios:

criação e edição

redirecionamento (apontamento)

Administração de FTP:

criação e edição de contas

Estatísticas de uso:

espaço em disco

banda

e-mails

sub-domínios

Informações sobre o Servidor:

status dos serviços

versões dos serviços

caminho do perl e do sendmail

- Também podemos Fazer Backup do site e Restaurar opções para backup completo, da pasta home ou dos bancos de dados do MySQL.

- Acesso via SSH

Um recurso muito útil para quem mexe com Linux. Com este podemos alterar o dono dos arquivos de um diretório, coisa que não se pode fazer pelo CPanel nem por um cliente de FTP.

Acesso Telnet shell

For security reasons, shell access is not enabled by default. In order to activate shell access on your account, you will need to fax or mail a copy of your driver's license, passport or other photo id to customer service.

Sua seção foi logada. Qualquer tentativa de copiar/excluir/ver arquivos que não são seus resultará na exclusão de sua conta!

Press to Start!

jta: ribafs.net 22

File Edit Terminal Help

Java Applet Window

```
.bash_logout                          .dns                public_html/
.bash_profile                         .emacs              tmp/
.bashrc                               etc/                .trash/
.contactemail                         .fantasticodata/    www@
.cpaddons/                            .htpasswds/
ribafsn@ribafs.net [~]# cd public_html
ribafsn@ribafs.net [~/public_html]# ls
./        .htaccess  joomla/        teste.php  _vti_cnf/      _vti_pvt/
../       images/    postinfo.html  tiago/     _vti_inf.html  _vti_txt/
cgi-bin/  index.php  _private/      _vti_bin/  _vti_log/      wordpress/
ribafsn@ribafs.net [~/public_html]# cd joomla
ribafsn@ribafs.net [~/public_html/joomla]# ls
./                     COPYRIGHT.php  index.php     offlinebar.php
../                    editor/        INSTALL.php   offline.php
administrator/         globals.php    language/     pathway.php
cache/                 help/          LICENSE.php   robots.txt
CHANGELOG.php          htaccess.txt   mainbody.php  templates/
components/            images/        mambots/
configuration.php*     includes/      media/
configuration.php-dist index2.php     modules/
ribafsn@ribafs.net [~/public_html/joomla]#
```

Connected to ribafs.net ssh online

Uso de clientes de FTP

Tutorial de Uso do gFTP (Linux)

Pode ser instalado pela opção Adicionar/Remover do Ubuntu ou por outro método em outra distro.

Este é um bom cliente de FTP, com os recursos necessários para se administrar sites remotamente.

Para a conexão digita-se o nome ou IP do do servidor remoto em Máquina, digita-se o nome do usuário em Usuário, digita-se a porta em Porta (apenas se for diferente da padrão, 21) e clica-se no botão de Conexão (dois computadores). Após clicar será solicitada a senha.

Uma boa dica é salvar os dados da conexão nos Marcadores (menu).

Como Alterar as Permissões de Arquivos e Pastas:

Clica-se sobre a pasta do lado direito (remoto) com o botão direito e seleciona-se Chmod...

No caso do Joomla, os diretórios precisam ter permissões como as acima para a instalação.

Tutorial de Uso do SmartFTP (Windows)

O SmartFTP é um bom cliente de FTP para Windows free.

Encontra-se em - http://www.smartftp.com/

Para fazer a conexão execute o programa:

Clique na caixa à direita de Address e selecione o protocolo FTP.

Digite à direita o nome ou IP do servidor, o Login e a Senha. Então clique na seta verde (Go) para conectar. Opcionalmente podemos clicar no último botão da imagem acima (New Remote Browser) para abrir o diálogo e entrar com os mesmos dados.

Então aparecerá a área remota do servidor à direita:

Alterando as Permissões de Arquivos

Após a conexão clique no arquivo ou pasta que deseja ter as permissões alteradas com o botão direito e Properties/CHMOD ou tecle F7 que abre o diálogo:

Veja que podemos tanto digitar o número quanto setar cada uma das permissões. Podemos também selecionar vários e alterar as permissões de todos ao mesmo tempo.

Sua própria empresa de hospedagem

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=6407

Ferramentas para Administração de Hospedagens

WinSCP

Ótimo software para acesso ao servidor via SSH (for Windows).

Ele abre um gerenciador de arquivos com dois painéis. À esquerda mostra os arquivos locais e à direita mostra os arquivos remotos.

Site oficial - http://winscp.net/eng/index.php

Existem versões para instalação e standalone.

putty

For windows, para acesso ao servidor via SSH.

Site oficial - http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Obs.: Caso possa, evite usar o FTP, prefira acesso via SSH que é mais seguro.

Krusader

Semelhante ao WinSCP mas este é somente for Linux.

Sua instalação depende do Konqueror (que deve ser instalado antes).

http://krusader.sourceforge.net/

SSH

Caso use Linux poderá acessar diretamente da console com o comando:

ssh usuario@dominio, exemplo: ssh joao@dominio.org

Caso esteja em um computador com conexão lenta, você pode se conectar ao servidor via SSH e fazer os downloads de lá mesmo usando o wget, que geralmente está instalado no servidor de hospedagem.

Fantastico

O software de administração de hospedagem mais popular em servidores Linux é o cPanel, que realmente tem muitos recursos úteis. O cPanel traz um utilitário chamado "Fantastico", que vem com inúmeros softwares prontos para instalar e cuja instalação é muito simplificada. Traz Joomla, Drupal, Xoops, Moodle e muitos outros. Varia de acordo com o servidor e a versão do cPanel.

Para usuários iniciantes é muito útil. Caso tenha boa experiência talvez seja preferível você mesmo fazer a instalação para ter um maior controle do que está acontecendo.

MySQL Quick Admin

Ferramenta para gerenciamento do MySQL.

Muito útil, especialmente se por algum motivo você esteja sem poder acessar seu site através do cPanel (firewall ou outro motivo).

Instale o MySQL Quick Admin em seu servidor, crie um banco para usos diversos e sempre que precisar crie as tabelas nesse banco com um prefixo.

Site oficial - http://www.mysqlquickadmin.com/

Hospedagem Grátis com Suporte a PHP e MySQL

- http://netfast.org/ (Esta tem scripts prontos para instalar com facilidade através do Fantastoco)
- http://www.phpnet.us/
- http://www.php0h.com/
- http://www.funpic.org/
- http://www.prophp.org/
- http://www.110mb.com/
- http://www.freeweb7.com

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Hospedagem_e_Dom%C3%ADnio/Hospedagem_de_Site_com_cPanel"
Page categories: PHP

Aplicativos em PHP/Apêndices/Segurança

Algumas Dicas sobre Segurança no Desenvolvimento de Aplicativos

Conteúdo

- 1 PHP Security Guide
- 2 Securing PHP: Step-by-Step
- 3 Apache 2 with SSL/TLS: Step-by-Step, Part 1 (Infocus)
- 4 Securing Apache 2: Step-by-Step
- 5 Securing MySQL: step-by-step (Infocus)
- 6 Securing Apache: Step-by-Step (Infocus)
- 7 Secure Your Apache With mod_security
- 8 Senhas
- 9 - Informações
- 10 - Código
- 11 Não Confiar em Variáveis Globais
- 12 Evite Falsos Uploads
- 13 Idéias Adicionais
- 14 Não armazene números de cartões de crédito
- 15 Não insira Conteúdo Sigiloso no raiz do Aplicativo
- 16 Muita atenção aos Serviços de Hospedagem
- 17 Evite ao máximo as funções
- 18 Além do Código (Um projeto de segurança forte)
- 19 Nunca inclua, requeira ou abra um arquivo cujo nome seja baseado em entrada do usuário, sem antes checar
- 20 Nunca execute consultas a bancos sem usar funções de escape
- 21 Restringe o diretório admin para utilizar senhas
- 22 Prevenindo Injeções SQL
- 23 Escrevendo Código Robusto em PHP
- 24 Formulários com Imagens anti-spam - Captcha
    - 24.1 PHP Captcha Security Images
    - 24.2 PHP Captcha
    - 24.3 iCaptcha - CAPTCHA validation asking questions about pictures
    - 24.4 Toughen Forms' Security with an Image
- 25 PHPSecInfo
- 26 Como tornar o PHP mais seguro
- 27 Convert mail to image (guard mail) from spam
- 28 Parser HTML inject
- 29 Referências

PHP Security Guide

Fundado em Janeiro de 2005, o PHP Security Consortium (PHPSC) é um grupo internacional formado por experts em PHP dedicados a promover práticas de segurança para a comunidade de PHP. Membros do PHPSC procuram educar desenvolvedores PHP sobre segurança através de uma variedade de recursos, incluindo documentação, ferramentas e padrões. Veja no linque abaixo o guia oferecido online pelo grupo:

http://phpsec.org/projects/guide/

Securing PHP: Step-by-Step

http://www.securityfocus.com/print/infocus/1706

Apache 2 with SSL/TLS: Step-by-Step, Part 1 (Infocus)

Artigo em 3 partes

http://www.securityfocus.com/infocus/1818

Securing Apache 2: Step-by-Step

http://www.securityfocus.com/infocus/1786

Securing MySQL: step-by-step (Infocus)

http://www.securityfocus.com/infocus/1726

Securing Apache: Step-by-Step (Infocus)

http://www.securityfocus.com/infocus/1694

Secure Your Apache With mod_security

http://www.howtoforge.com/apache_mod_security

Senhas

Ataques contra senhas normalmente usam o método da força bruta (brute force), portanto evite senhas simples e que constem de dicionários.

Evite senhas do tipo:

- Somente numéticas ou somente caracteres

- Seu nome, nomes de filhos, número de identidade, data de nacimento, CEP ou outro constante de dicionários

- Senhas com poucos caracteres

- Não use a mesma senha para todos os servidores

Senha recomendada:

- Com 15 ou mais caracteres

- Misture letras, algarismos e caracteres especiais, maiúsculas e minúsculas

- Muito importante: use uma senha que você possa se lembrar

- Informações

Cuidado com o Acesso em Lan houses

Segundo a PNAD 2005 (IBGE) 18,6% das residências brasileiras têm computadores e destes apenas 13,7% com acesso à Internet. Muita gente então acessa a Internet em Lan hauses.

Os especialistas em segurança recomendam não efetuar transações bancárias ou qualquer outro tipo onde sejam requeridos dados pessoais, como CPF, cartão de crédito, RG, etc.

Existem diversas formas de se monitorar os passos de alguém na Internet (programas mal intencionados) e até mesmo as câmaras existentes nas lan houses, se em mãos erradas podem representar perigo para os usuários.

Cuidado com Notebooks

Os notebooks também estão na mira dos criminosos da Internet. Evite dados sigilosos nos mesmos ou use uma boa criptografia para proteger seus dados. Use senhas fortes para o acesso ao notebook e faça backup regularmente de suas informações.

Instale e tenha sempre atualizados: bom antivirus, antispy e firewall.

Esteja atento para as atualizações do sistema operacional e de todos os softwares importantes.

- Código

- Ofereça uma quantidade fixa (3) de tentativas de login. Após as 3 o login deve ser desabilitado,

por segurança.

Segurança em Aplicativos PHP

Este texto é formado pela tradução de partes de alguns textos em inglês (vide Referências) e de alguns exemplos de código e recomendações que adicionei.

Em primeiro lugar devemos atentar para uma boa análise e projeto da aplicação. Da qualidade destes depende a qualidade da aplicação. Então devemos planejar o banco de dados cuidadosamente: tabelas, relacionamentos, campos, tipos de dados, etc. Mais importantes ainda em termos de segurança é a criação de usuários que tenham acesso somente ao aplicativo e com privilégios somente para suas operações com nomes e senhas seguras, como também usuários do banco com respectivas permissões e permissões do sistema operacional.

As aplicações Web contam com formas populares de acesso global a dados, a serviços e a produtos. Enquanto este acesso global é uma das grandes vantagens da Web, qualquer regra de segurança nesses aplicativos também é globalmente exposta e freqüentemente explorada. É muito fácil escrever aplicações que contém regras de segurança. Vide aplicações famosas como phpMyAdmin, PHPShop e FreeTrade.

Algumas Recomendações a favor da Segurança

- Evitar uso de Variáveis quando acessando Arquivos

Cuidado com as funções:

- readfile

- fopen

- file

- include

- require

Caso decida assim mesmo utilizar, tome precauções. Uma boa precaução é que o valor das variáveis seja definido com o uso da função "define", garantindo que seu conteúdo seja conhecido e testado.

- Checar os nomes dos arquivos em uma lista de nomes válidos. Veja um exemplo: $valid_pages = array("umapagina.php" => "", "outra.php" => "", "mais.php" => "");

if (!isset($valid_pages[$page])) { // Aborte o script //Você deve provavelmente escrever uma mensagem de log aqui também die("Requisição inválida"); }

- Caso realmente precise usar variáveis de um browser, cheque os valores das variáveis usando um código como o seguinte: if (!(eregi("^[a-z_./]*$", $page) && !eregi("\\.\\.", $page))) { // Abortar o script //Você deve provavelmente escrever uma mensagem de log aqui também die("Requisição inválida "); }

- Use as configurações de variáveis “allow_url” e “open_basedir” para limitar as localizações de onde os arquivos podem ser abertos.

Utilizar Caracteres de Escape em Instruções SQL

// Usar a função para testar a existência de registro if (record_exists($query)) { echo "Acesso ragantido"; } else { echo "Acesso negado"; }

O uso da diretriz “magic_quotes_gpc” setada para On no php.ini insere caracteres de escape nas Super Globais $_POST, $_GET e COOKIES. Veja o exemplo abaixo do site oficial:

<?php echo get_magic_quotes_gpc(); // 1 echo $_POST['lastname']; // O\'reilly echo addslashes($_POST['lastname']); // O\\\'reilly

if (!get_magic_quotes_gpc()) {

```
  $lastname = addslashes($_POST['lastname']);
```

} else {

```
  $lastname = $_POST['lastname'];
```

}

echo $lastname; // O\'reilly $sql = "INSERT INTO lastnames (lastname) VALUES ('$lastname')"; ?
>

Use addslashes e stripslashes, caso esteja usando variáveis globais (register_globals = On) e não esteja usando magic_quotes_gpc como no exemplo abaixo (adiciona antes de inserir no banco e remove antes de exibir na tela):

// Recebendo do Form

$thisCodigo_curso = addslashes($_REQUEST['thisCodigo_cursoField']); $thisNome = addslashes($_REQUEST['thisNomeField']);

$sqlQuery = "INSERT INTO curso (codigo_curso , nome) VALUES ('$thisCodigo_curso' , '$thisNome')";

// Enviando para a tela

eval('function cndstrips($str) {

```
 return '.(get_magic_quotes_gpc()?'stripslashes($str)':'$str').';
```

}');

Obs.: O uso simultâneo de magic_quotes_gpc = On com addslashes e stripslashes gera problemas, assim como não podemos usar register_globals = On simultaneamente com as Super Globais $_POST, $_GET.

Caso esteja usando variáveis que espera números para seu conteúdo nas instruções SQL, esteja seguro de que realmente contém números. Existem diversas funções em PHP incluindo sprintf, ereg e is_long para realizar a checagem. Também podemos utilizar o JavaScript para checar as entradas logo no formulário.

Não Confiar em Variáveis Globais

Se register_globals = On no php.ini, então o PHP criará variáveis globais para cada requisição GET, POST e variáveis Cookie.

Preste bastante atenção nas seguintes áreas:

- Código de checagem de autenticação e permissão

- Uso de variáveis antes de serem inicializadas. (Podemos ajustar error_reporting para ser alertados sempre que se usar variáveis não inicializadas.

- Uso de variáveis designadas para ser usadas por requisições GET ou POST.

Veja da documentação oficial:

Exemplos error_reporting()

<?php

// Desativa o relatório de todos os erros error_reporting(0);

// Reporta erros simples error_reporting(E_ERROR | E_WARNING | E_PARSE);

// Reportar E_NOTICE pode ser bom também (para reportar variáveis não iniciadas // ou eros de digitação em nomes de variáveis ...) error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);

// Reportar todos os erros exceto E_NOTICE // Este é o valor padrão no php.ini error_reporting(E_ALL ^ E_NOTICE);

// Reporta todos os erros (bitwise 63 deve ser usado no PHP 3) error_reporting(E_ALL);

// O mesmo que error_reporting(E_ALL); ini_set('error_reporting', E_ALL);

?>

Possíveis Correções e Melhorias

Desabilitar “register_globals” no php.ini. Após esta mudança somente podemos acessar entradas de formulários usando $_POST ou $_GET. Fica mais trabalhoso um pouco mas bem mais seguro (vale a pena).

Cuidado com o recurso “Esqueceu a Senha” de formulários de Login.

Escreva código para inicializar todas as variáveis globais.

Evite Falsos Uploads

Examine todos os scripts que respondem a upload de arquivos.

Use as funções is_uploaded_file e move_upload_file que permitem ao programador estar seguro de que está trabalhando com os devidos arquivos enviados.

Caso não tenha certeza de estar rodando em uma versão atual do PHP, configure upload_tmp_dir que executa checagem de entrada que o arquivo que estamos trabalhando está neste diretório.

Idéias Adicionais

- Encripte ou use hashes de senhas quando armazenando. A função md5 é útil para isso.

Exemplo retirado do www.phpbrasil.com:

//BY ADEMIR BATISTA PEREIRA $senha = "ribafs"; $resultado = md5($senha); $resultado2 = bin2hex($senha); if($senha == ""){ $texto = ""; } else{ $texto = "
Criptografia tipo A para: $senha é $resultado"; echo $texto; $texto2 = "
Criptografia tipo B para: $senha é $resultado2"; echo $texto2; }

Outro exemplo do mesmo site:

<?php //***** enc - Encriptador de String //***** Autor: Ricardo Antonio Duarte - ricardo NOSPAM banhado.com

function enc($string){

```
    if((isset($string)) && (is_string($string))){
        $enc_string = base64_encode($string);
        $enc_string = str_replace("=","",$enc_string);
        $enc_string = strrev($enc_string);
        $md5 = md5($string);
        $enc_string = substr($md5,0,3).$enc_string.substr($md5,-3);
    }else{
        $enc_string = "Parâmetro incorreto ou inexistente!";
    }
    return $enc_string;
```

} //***** Fim do enc

//***** des - Desencriptador de String //***** Autor: Ricardo Antonio Duarte - ricardo NOSPAM banhado.com

function des($string){

```
    if((isset($string)) && (is_string($string))){
        $ini = substr($string,0,3);
        $end = substr($string,-3);
```

```
        $des_string = substr($string,0,-3);
        $des_string = substr($des_string,3);
        $des_string = strrev($des_string);
        $des_string = base64_decode($des_string);
        $md5 = md5($des_string);
        $ver = substr($md5,0,3).substr($md5,-3);
        if($ver != $ini.$end){
            $des_string = "Erro na desencriptação!";
        }
    }else{
        $des_string = "Parâmetro incorreto ou inexistente!";
    }
    return $des_string;
```

} //***** Fim do des

echo "Enc: ".enc("ribafs")."
";

echo "Des: ".des("53czZWYilmc1d1");

?>

Não armazene números de cartões de crédito

Force senhas seguras. No mínimo exija senhas com 8 caracteres de tamanho e que contenham algum caractere não alfanumérico.

Avoid SQL injection

This is a function which will format the passed string depending of it's specified to be a number or a string, in order to avoid problems with SQL injections in scripts.

Type: code fragment

Version:

Requires:

Added by: bto (email author)

Entered: 19/08/2004

Last modified: 08/12/2003

Rating - (fewer than 3 votes)

Views 3024

<?php function ToDBString($string, $link, $isNumber=false) {

```
    //If $isNumber==true we are specting a number
    if($isNumber)
    {
        //A correct number must be composed of:
        // - Zero or more integers followed by a decimal point and one or more
integers (i.e.: .9 (0.9) or 9.9)
        // - One or more integers followed by a decimal point. (i.e.: 9. (9.0))
```

```
        // - One or more integers (i.e.: 999)
        if(preg_match("/^d*[.,']d+|d+[.,']|d+$/A", $string))
        //If it's a correct number we change the colon, quote or point ("'", ","
or ".") by a decimal piont.
            return preg_replace( array(
                                        "/^(d+)[.,']$/"     , //9.
                                        "/^(d*)[.,'](d+)$/"  //.9 or 9.9
                                       ),
                                 array(
                                        "\1."               ,
                                        "\1.\2"
                                       )
                                 , $string);
        else
        //If it's not a correct number we show ERROR
            die("ERROR: Not a number"".$string.""");
    }
    else
     //If $string is a string ($isNumber==false) we return "'$string'"
     // correctly escaped (in this version I also strip HTML tags and modify some
things in the string, change it if you wish).
     return
"'".mysql_real_escape_string(htmlentities(strtoupper(trim(strip_tags($string))))
, $link)."'";

} ?>
```

Example

$link=mysql_db_connect("HOST", "USER", "PASSWORD");

$foo=ToDBString($_POST["string"], $link); $bar=ToDBString($_POST["number"], $link, true);

$result=mysql_db_query("DATABASE", "SELECT * FROM secret WHERE foo LIKE $foo AND bar=$bar", $link);

//If $_POST["foo"] or $_POST["bar"] are a string of this kind: " *OR 1=1" and we don't use ToDBString we will show all the info of the table!!!!*

- Usar sempre extensões tipo PHP em todos os scripts.

Nunca usar extensões .inc, .class, etc. Podemos usar assim: nome.inc.php.

Arquivos com extensões .inc, .class e similares ao serem abertos no browser exibem todo o seu conteúdo, inclusive senhas de banco, trechos de código PHP e outras, já script .php, antes de serem abertos no browser são processados pelo servidor web e só chega ao browser o resultado em HTML.

Não insira Conteúdo Sigiloso no raiz do Aplicativo

Imagens de proteção de senha, documentos e outras imagens devem ficar fora do raiz do aplicativo.

Alternativamente armazene no banco de dados.

Proteger os diretórios usando características do Apache, como arquivos .htaccess, que previnem o acesso direto ao conteúdo dos diretórios.

Muita atenção aos Serviços de Hospedagem

Estes servidores compartilham suas máquinas com diversos usuários, que têm acesso aos arquivos. Como também podem criar um arquivo de session (que armazena em /tmp por default), que pode conter maliciosos users e senhas para bypassar sua autenticação.

Verifique sempre as informações do PHP (phpinfo()) do servidor, caso use esses serviços.

Idealmente use um servidor dedicado ao invés.

Assegure-se de que o servidor ative safe_mode no php.ini.

As permissões dos arquivos são outro ponto importante. Devem somente ser lidos pelo web Server. No UNIX use algo como 711.

Use sempre validações para garantir que os usuários realmente entram com informações válidas. is_long, is_numeric, etc.

Fuja ou Evite Entrada de Usuários quando Construindo Comandos de Strings Funções como exec e eval são muito flexíveis mas requerem muito cuidado, pois os usuários podem entrar com comandos inesperados.

Evite ao máximo as funções

- eval
- preg_replace (quando usada com /e deve interpretar parâmetros como código PHP)
- exec
- passthru
- system
- popen

- (pode ser usado para executar comandos)

Além do Código (Um projeto de segurança forte)

Projeto de Aplicação Segura

- Considere o uso do HTTPS para encriptar transmissões.

- Considere restringir acesso aos diretórios usando .htaccess do Apache. Checar através das variáveis de ambiente do PHP, como $REMOTE_ADDR

- Usar pacotes de segurança existentes, como o PHPLib.

Todas as linguagens tem seus pontos fracos, mas tomando diversos cuidados preventivos e atentando para as boas regras de segurança, estes pontos podem ser protegidos. Seguindo os passos mostrados aqui podemos desenvolver um código mais seguro que o usual.

Um dos mais importantes conceitos a se ter em mente em termos de segurança é o de nunca confiar que o usuário irá digitar exatamente o que se espera que ele digite.

Nunca inclua, requeira ou abra um arquivo cujo nome seja baseado em entrada do usuário, sem antes checar

- Seja muito cuidados quando usando “register_globals = On”.

Isso foi feito para tornar o uso do PHP algo fácil, o que realmente aconteceu. Mas em contrapartida trouxe sérios problemas de segurança. A partir da versão 4.2.0 este parâmetro já vem setado como Off por default. Neste caso para pegar o valor de variáveis lançadas pelo formulário devemos utilizar as superglobais $_POST, $_GET, $_REQUEST, $_COOKIE. Ou $_SESSION.

O recomendado é que se trabalhe com “register_globals = Off”.

Usando “error_reporting = E_ALL” no php.ini recebemos uma notificação sempre que tentarmos chamar variáveis que ainda não tenham sido definida. Sabemos que o PHP não exige a definição de variáveis, mas recomenda-se que nos acostumemos a definir todas as variáveis e inclusive a iniciá-las, em termos de segurança.

Nunca execute consultas a bancos sem usar funções de escape

O PHP traz ativa por default, uma proteção contra a entrada de caracteres especiais nos formulários, que é o "magic_quotes_gpc = On".

- Nunca confie em dados de fontes externas.

- Toda entrada de usuário deve ser validada e formatada para garantir a segurança.

Protegendo Arquivos e Diretórios com .htaccess via Apache

- Para ativar o .htaccess em todo o servidor web, edite o httpd.conf e adicione:

<Directory />

```
Options FollowSymLinks Indexes
AllowOverride AuthConfig
```

</Directory>

- Então criamos o arquivo .htaccess existente no diretório raiz e adicionamos tags de acordo com nossos propósitos. Exemplos:

- Restringindo o acesso por IP/Host

1. Deixa a Intranet acessar

Order allow,deny allow from 192.168.0. deny from all

Ou

1. Deixa todo mundo acessar, menos o IP 192.168.0.25

Order deny,allow deny from 192.168.0.25 allow from all

- Restringindo o acesso por user/senha

$ mkdir /etc/httpd/auth

$ cd /etc/httpd/auth

$ htpasswd -c acesso hugo

New password:

Re-type new password:

Adding password for user hugo

$ htpasswd acesso eitch

New password:

Re-type new password:

Adding password for user eitch

$ htpasswd acesso sakura

New password:

Re-type new password:

Adding password for user sakura

- Agora criar o .htaccess:

AuthName "Acesso Restrito à Usuários"

AuthType Basic

AuthUserFile /etc/httpd/auth/acesso

require valid-user

AuthUserFile /etc/httpd/auth/acesso – onde estão as senhas e users.

- Opções para arquivos e diretórios específicos:

1. Restringe o arquivo_secreto.html somente para o IP 192.168.0.30

<Files arquivo_secreto.html>

```
Order allow,Deny
Allow from 192.168.0.30
Deny from all
```

</Files>

Restringe o diretório admin para utilizar senhas

<Directory /admin>

```
AuthName "Acesso Restrito à Usuários"
AuthType Basic
AuthUserFile /etc/httpd/auth/acesso
AuthGroupFile /etc/httpd/auth/grupos
require group admin
```

</Directory>

1. Nega o acesso dos clientes ao .htaccess (bom colocar no httpd.conf)
2. - Vem com a configuração padrão do Apache

<Files ~ "^\.ht">

```
   Order allow,deny
   Deny from all
```

</Files>

Prevenindo Injeções SQL

Uma Função para prevenir tais injeções.

Esta é uma função que deve formatar a string passada dependendo de se foi especificado um número ou uma string, para evitar o problema com injeções SQL em scripts.

Autor: bto (no site

<?php function ToDBString($string, $link, $isNumber=false) {

```
   //If $isNumber==true we are specting a number
   if($isNumber) {
   //A correct number must be composed of:
   //Zero or more integers followed by a decimal point and one or more integers
(i.e.: .9 (0.9) or 9.9)
       // - One or more integers followed by a decimal point. (i.e.: 9. (9.0))
       // - One or more integers (i.e.: 999)

       if(preg_match("/^d*[.,']d+|d+[.,']|d+$/A", $string))
   //If it's a correct number we change the colon, quote or point ("'", "," or
".") by a decimal piont.
           return preg_replace( array(
                                      "/^(d+)[.,']$/"     , //9.
                                      "/^(d*)[.,'](d+)$/"  //.9 or 9.9
                                     ),
                                array(
                                      "\1."               ,
                                      "\1.\2"
                                     )
                                , $string);
       else
       //If it's not a correct number we show ERROR
           die("ERROR: Not a number"".$string.""");
   }
   else
    // If $string is a string ($isNumber==false) we return "'$string'"
    // correctly escaped (in this version I also strip HTML tags and modify some
things in the string, change it if you wish).
    return
"'".mysql_real_escape_string(htmlentities(strtoupper(trim(strip_tags($string))))
, $link)."'";
```

} ?>

Example

$link=mysql_db_connect("HOST", "USER", "PASSWORD");

$foo=ToDBString($_POST["string"], $link); $bar=ToDBString($_POST["number"], $link, true);

$result=mysql_db_query("DATABASE", "SELECT * FROM secret WHERE foo LIKE $foo AND bar=$bar", $link);

//If $_POST["foo"] or $_POST["bar"] are a string of this kind: " *OR 1=1" and we don't use ToDBString we will show all the info of the table!!!!*

Escrevendo Código Robusto em PHP

Formulários com Imagens anti-spam - Captcha

PHP Captcha Security Images

Sistema de imagens para controle de spam em PHP

http://www.white-hat-web-design.co.uk/articles/php-captcha.php

Link para download - http://www.white-hat-web-design.co.uk/articles/captcha.zip

PHP Captcha

http://milki.erphesfurt.de/captcha/

Download - http://milki.erphesfurt.de/captcha/captcha-2.0.tgz

iCaptcha - CAPTCHA validation asking questions about pictures

http://www.phpclasses.org/browse/package/3960.html

Toughen Forms' Security with an Image

Artigo detalhado descrevendo como criar um form seguro tipo captcha

http://www.sitepoint.com/print/toughen-forms-security-image

PHPSecInfo

Utilitário semelhante à função phpinfo(), que mostra as falhas de segurança do PHP e aponta sugestões para correção.

http://phpsec.org/projects/phpsecinfo/

Como tornar o PHP mais seguro

http://www.jorgeoliveira.com/2006/04/03/como-tornar-o-php-mais-seguro/

Convert mail to image (guard mail) from spam

http://snippets.dzone.com/posts/show/4283

```
<?php

if(empty($_GET['sid'])){
        //write html for image
        $mail='example@domain.com';
    $text=base64_encode(serialize($mail));
        echo '<img src="imgmail.php?sid='.$text.'" /><br/>';
}else{
        //show coded mail image
        $text=$_GET['sid'];
        header("Content-type: image/gif");
        echo mail_to_image($text);
}

function mail_to_image($ctext='no mail'){
        $text=unserialize(base64_decode($ctext));
    $size=strlen($text)*8;
    $im = @imagecreate($size, 20) or die("Cannot Initialize new GD image
stream");
    $background_color = imagecolorallocate($im, 255, 255, 255);
    $text_color = imagecolorallocate($im, 0, 0, 0);
    imagestring($im,3, 5, 5,  $text , $text_color);
    return imagegif($im);
}
?>
```

Parser HTML inject

Parser HTML inject. Bom para quando temos a possibilidade de receber em uma textarea código HTML ou JavaScript malicioso.

Código fuente / Source code :

```
function parsearHTMLInjectado($texto)
{
        return nl2br( htmlentities($texto) );
}
```

Fonte: http://snippets.dzone.com/posts/show/4347

Referências

1 – http://www.onlamp.com/lpt/a/3305 - Ten Security Checks for PHP

2 - http://www.onlamp.com/lpt/a/4045 - PHP Security

3 - http://www.devshed.com/c/a/PHP/PHP-Security-Mistakes/ - PHP Security Mistakes

4 - http://www.devin.com.br/eitch/htaccess/ - Uso e Segurança com o .htaccess

5 - http://www.securiteam.com/securityreviews/5DP0N1P76E.html - SQL injeções

6 - http://www.zend.com/codex.php?id=1405&single=1 - SQL injeções

7 – http://www.imasters.com.br/imprimir.php?cn=292&cc=44

8 – http://www.imasters.com.br/imprimir.php?cn=293&cc=44

9 – http://www.imasters.com.br/imprimir.php?cn=319&cc=44

10 - Escrevendo o código seguro de PHP

11 - - Writing Secure PHP

12 - - Vulnerabilidade em formulário PHP

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Seguran%C3%A7a"
Page categories: PHP

Aplicativos em PHP/Apêndices/Documentação

De Wikibooks

< Aplicativos em PHP | Apêndices

Documentando o Desenvolvimento

A documentação do código, do projeto, do banco e de todas as fases de um aplicativo é algo importante e mais quanto maior e complexo for o aplicativo.

Existe um utilitário que foi inspirado no JavaDoc, chamado PHPDoc ou PHPDocumentor, encontrado em - http://www.phpdoc.org/, que ajuda na documentação do código PHP.

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Documenta%C3%A7%C3%A3o"
Page categories: PHP

Aplicativos em PHP/Apêndices/Informações diversas

De Wikibooks

< Aplicativos em PHP | Apêndices

Conteúdo

- 1 Informações Diversas
- 2 Caching your pages with PhP
- 3 Conectar com MS Access
  - 3.1 Connect with Access
- 4 Links da Área

Informações Diversas

Caching your pages with PhP

http://www.webdigity.com/index.php?action=tutorial;code=5

Conectar com MS Access

Connect with Access

```
$pathDB = str_replace("/", "\\", $_SERVER["DOCUMENT_ROOT"]) .
"\\directory1\\directory2\\bdd.mdb";

if(!file_exists($pathDB))
{
echo "!!! Base de datos no encontrada ".$pathDB;
exit;
}

$conexion = odbc_connect("DRIVER={Microsoft Access Driver (*.mdb)};
DBQ=".$pathDB, "ADODB.Connection", "", "SQL_CUR_USE_ODBC");
$sql="select * from tabla where 1";
$resultado=odbc_exec($conexion,$sql);
if($resultado)
{
        while($fila=odbc_fetch_array($resultado))
        {
```

```
        echo $fila['campo1']."<br />";
        echo $fila['campo2']."<br />";
        }
odbc_close_all();
}
```

Fonte: http://snippets.dzone.com/posts/show/4345

Links da Área

- PHP.net - http://www.php.net
- PHPBrasil - http://www.phpbrasil.com
- Artigos, tutoriais e dicas de PHP - http://www.htmlstaff.org/secao.php?id=18
- PHP Classes – http://www.phpclassess.org
- HotScripts (PHP, JavaScript e outros) – http://www.hotscripts.com
- Planet Source Code – http://www.planet-source-code.com
- Revista PHP - http://www.revistaphp.com.br/
- PHP Magazine - http://www.phpmagazine.com.br/
- Desenvolvimento Web – http://ribafs.net
- Aplicativos em PHP – http://pt.wikibooks.org/wiki/Aplicativos_em_PHP
- PostgreSQL Prático - http://pt.wikibooks.org/wiki/PostgreSQL_Prático
- Leitor de RSS do Google – http://reader.google.com
- Repositório de Projetos SourceForge – http://www.sourceforge.net
- Repositório de Projetos Código Livre – http://www.codigolivre.org.br
- Repositório de Projetos (Open Sources) da Microsoft - http://www.codeplex.com/
- Repositório de Projetos do Google - http://code.google.com/hosting/ (requer conta do Gmail)
- Pesquisa de código-fonte público - http://www.google.com/codesearch
- Repositório de Projetos do FreshMeat – http://www.freshmeat.net
- Todos os projetos do Google - http://labs.google.com/
- Páginas HTML gratuitas no Google – http://googlepages.com
- Blog Gratuito - http://wordpress.com/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Informa%C3%A7%C3%B5es_diversas"
Page categories: PHP

Aplicativos em PHP/Apêndices/Oportunidade de Trabalho

De Wikibooks

< Aplicativos em PHP | Apêndices

Conteúdo

- 1 16.7 - Oportunidade de Trabalho
- 2 Produtividade Pessoal na busca de Emprego
- 3 K.I.S.S.
- 4 3 dicas de produtividade para desenvolvedores web
- 5 Organize-se, mas também desorganize-se!
- 6 10 Dicas de Produtividade
- 7 Autoconhecimento e Efetividade
- 8 A Usabilidade e o Espelho
- 9 Produtividade
- 10 Organização = Efetividade Máxima
- 11 | modo de vida | Produtividade
- 12 Como otimizar seu tempo
- 13 Os Sete Segredos dos que Nunca estão Desempregados
- 14 Guias de Profissões
- 15 12 técnicas de marketing online
- 16 Obter Renda. Vender um Produto ou Serviço
- 17 10 dicas para ganhar dinheiro de verdade com seu site ou blog usando o Google Adsense (parte 2 de 2)
- 18 GTD: 4 dicas simples para facilitar a adoção deste método de produtividade pessoal

16.7 - Oportunidade de Trabalho

1. Elaboração de Currículos
2. Técnicas de Entrevistas
3. Preços de Serviços

Produtividade Pessoal na busca de Emprego

http://laboro.blogspot.com/2007/07/produtividade-pessoal-na-busca-de.html

K.I.S.S.

Uma tradução livre - Mantenha tudo simples seu simples. Recomendações para que não compliquemos as coisas, mas ao contrário as mantenhamos o mais simples possível.

http://oidiota.wordpress.com/2007/07/05/kiss/

3 dicas de produtividade para desenvolvedores web

http://blog.klaus.pro.br/ler/web/2007/3-dicas-de-produtividade-para-desenvolvedores-web/33/index.html

Organize-se, mas também desorganize-se!

http://floresilveira.blogspot.com/2007/07/organize-se-mas-tambm-desorganize-se.html

10 Dicas de Produtividade

http://gabrielgalvao.com/?p=104

Autoconhecimento e Efetividade

http://jotaka.wordpress.com/2007/07/12/autoconhecimento-e-efetividade/

A Usabilidade e o Espelho

http://deloyola.wordpress.com/2007/07/11/a-usabilidade-e-o-espelho/

Produtividade

Artigo sobre como melhorar nossa produtividade no trabalho:

http://www.palpitedigital.com.br/wp/2007/07/18/produtividade/

Organização = Efetividade Máxima

Boas recomendações para otimizarmos nosso dia-a-dia

http://www.guiaware.com/blogosfera/organizacao-efetividade-maxima

| modo de vida | Produtividade

http://saladaurbana.blogspot.com/2007/07/modo-de-vida-produtividade-e.html

Como otimizar seu tempo

http://cottonsoul.blogspot.com/2007/07/como-otimizar-seu-tempo.html

Os Sete Segredos dos que Nunca estão Desempregados

Maria Jesús Ribas em seu artigo "Como encontrar um emprego e se fortalecer no mercado de trabalho" para o Yahoo! Notícias aborda tópicos do livro "Os Sete Segredos dos que Nunca estão Desempregados" dos especialistas franceses Nicolás Buisson e Charles Henri Dumon. Veja alguns dos tópicos que achei interessantes e que podem ajudar cada um de nós:

- O dado negativo é que o sucesso profissional não está garantido a ninguém.

- O marketing pessoal, o famoso "vender o seu próprio peixe", não tem grandes segredos: consiste em estudar os pontos fortes e fracos de sua carreira para tirar proveito de suas virtudes e corrigir seus defeitos.

- Os que nunca estão desempregados - são pessoas que entendem como evolui o mercado de trabalho e sabem em qual setor ele está fortalecido. Para isso é preciso analisar seu entorno profissional, avaliando riscos e oportunidades.

Além disso estes trabalhadores conhecem bem a si mesmos, avaliando seu desempenho dentro de

uma companhia, a utilidade de sua produção, sua necessidade dentro de uma companhia, a utilidade de sua produção, sua necessidade dentro da empresa, o relacionamento com seus chefes e a avaliação externa do seu trabalho.

Alguém pode estar em uma companhia perfeita, em um cargo perfeito, mas o ambiente de trabalho ou o relacionamento com as pessoas a sua volta não ser tão bom. Neste caso é preferível a busca por outro projeto.

- Pense como uma empresa - Este conceito precisa que cada profissional se veja como uma mini-empresa que comercializa seu valor dentro da companhia. Uma pessoa passiva, que se importa apenas com seu salário e que pensa que tem um trabalho a ser feito e nada mais, precisa rever suas atitudes. O pensamento correto é o de se dedicar ao máximo para conquistar uma escalada profissional dentro de sua empresa, ocupando cada vez mais cargos importantes.

- Aumente seu nível de empregabilidade - Em vez de esperar que a companhia lhe dê uma maior formação e te presenteie com promoções, busque se especializar por conta própria. Para isso comunique aos demais o trabalho que está fazendo, sem fazer política de si mesmo e sem pisar nos outros. É preciso ter uma rede de contatos importante, tanto dentro da empresa quanto fora dela. Deve-se estudar o que pode ser melhorado, assim como suas fraquezas e não parar de "se vender" em momento algum, para que seus superiores pensem em você quando eles precisarem de alguém qualificado.

Guias de Profissões

Muitas informações úteis para quem está ainda indeciso sobre a profissão a seguir ou para quem precisa de mais informações sobre profissões:

http://www.portaldeensino.com.br/profissoes.php

http://www.guiadasprofissoes.com.br/

http://www.brasilescola.com/guiadeprofissao/

http://www3.catho.com.br/guia/busca_alfabetica.php

12 técnicas de marketing online

http://www.criarweb.com/artigos/808.php

Obter Renda. Vender um Produto ou Serviço

http://www.criarweb.com/artigos/809.php

10 dicas para ganhar dinheiro de verdade com seu site ou blog usando o Google Adsense (parte 2 de 2)

http://www.efetividade.net/2006/10/17/10-dicas-para-ganhar-dinheiro-de-verdade-com-seu-site-ou-blog-usando-o-google-adsense-parte-2-de-2/

GTD: 4 dicas simples para facilitar a adoção deste método de produtividade pessoal

http://www.efetividade.net/2007/04/01/gtd-4-dicas-simples-para-facilitar-a-adocao-deste-metodo-de-produtividade-pessoal/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Oportunidade_de_Trabalho"
Page categories: PHP

Aplicativos em PHP/Apêndices/Oportunidade de Trabalho/Elaboração de Currículos

De Wikibooks

< Aplicativos em PHP | Apêndices | Oportunidade de Trabalho

Conteúdo

- 1 Elaboração de Currículos
- 2 Valorizando o currículo: como conseguir o emprego dos seus sonhos na área de Informática
- 3 Revolução etc: Como escrever e sustentar um bom currículo para vagas de webdesigner
- 4 Como escrever (e sustentar) um bom currículo para vagas de web designer
- 5 10 dicas de conteúdo para criar um modelo de curriculum caprichado
- 6 10 dicas de visual e formatação para criar um modelo de curriculum caprichado
- 7 Dicas Para Ajudar a Escrever e a Falar Melhor

Elaboração de Currículos

Deve ser conciso, resumido e preciso, para evitar que não seja lido.

Deve ser bem objetivo para atender ao objetivo solicitado.

Fale de sua experiência profissional e estudo.

Demonstre com segurança que de fato é o profissional de que a empresa precisa.

Seja convincente.

Valorizando o currículo: como conseguir o emprego dos seus sonhos na área de Informática

http://www.efetividade.net/2007/01/28/valorizando-o-curriculo-como-conseguir-o-emprego-dos-seus-sonhos-na-area-de-informatica/

Revolução etc: Como escrever e sustentar um bom currículo para vagas de webdesigner

http://www.efetividade.net/2007/01/09/revolucao-etc-como-escrever-e-sustentar-um-bom-curriculo-para-vagas-de-webdesigner/

Como escrever (e sustentar) um bom currículo para vagas de web designer

http://www.revolucao.etc.br/archives/como-escrever-e-sustentar-um-bom-curriculo-para-vagas-de-web-designer/

10 dicas de conteúdo para criar um modelo de curriculum caprichado

http://www.efetividade.net/2006/11/10/10-dicas-de-conteudo-para-criar-um-modelo-de-curriculum-caprichado/

10 dicas de visual e formatação para criar um modelo de curriculum caprichado

http://www.efetividade.net/2006/11/15/10-dicas-de-visual-e-formatacao-para-criar-um-modelo-de-

curriculum-caprichado/

Dicas Para Ajudar a Escrever e a Falar Melhor

http://www.geocities.com/ribafsindex/Escrever.htm

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Oportunidade_de_Trabalho/Elabora%C3%A7%C3%A3o_de_Curr%C3%ADculos"
Page categories: PHP

Aplicativos em PHP/Apêndices/Oportunidade de Trabalho/Técnicas de Entrevistas

De Wikibooks

< Aplicativos em PHP | Apêndices | Oportunidade de Trabalho

Técnicas de Entrevistas

É hora de mostrar que de fato conta com as habilidades relatadas no currículo.

Fale com segurança e dê exemplos do que faz e deseja fazer.

Não aborde assuntos que não domina mas se for solicitado ao do gênero fale que não conheçe mas que tem facilidade e boa vontade para aprender.

Procure ser inovador e criativo.

Seja organizado.

Procure conhecer a empresa e traga soluções que agregem valor para a mesma.

"Nunca haverá um fim no aprendizado, a não ser que acreditemos que o conhecimento que já temos é suficiente para nós."

"O que nos fará mais desenvolvidos não é o conhecimento ao qual teremos acesso e sim o que faremos para aprimorar a humanidade com todo esse conhecimento." Frederick Motero

Entrevista de emprego: como se sair bem - parte 1: antes da entrevista

http://www.efetividade.net/2006/12/13/entrevista-de-emprego-preparacao/

Entrevista de emprego, parte 2: mais 10 dicas para se sair bem DURANTE a entrevista

http://www.efetividade.net/2007/02/05/entrevista-de-emprego-parte-2-10-dicas-para-se-sair-bem-durante-a-entrevista/

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Oportunidade_de_Trabalho/T%C3%A9cnicas_de_Entrevistas"
Page categories: PHP

Aplicativos em PHP/Apêndices/Oportunidade de Trabalho/Preços de Serviços

De Wikibooks

< Aplicativos em PHP | Apêndices | Oportunidade de Trabalho

16.7.3 - Preços de Serviços

Uma boa idéia é procurar os sindicatos e cooperativas da área especialmente em sua região ou cidade.

Eles sempre oferecem uma tabela de preços dos serviços da categoria.

Aqui segue o endereço do **Sindicato dos Trabalhadores em Processamento de Dados e Informática no Ceará**

http://www.sindpdce.org.br/

Documento da convenção do referido sindicato, contendo tabelas com salários bases das categorias:

http://www.sindpdce.org.br/download/convencaocoletiva/cct_sindpd_seacec_jan_2006.pdf

Adicione o do seu estado e de outros estados aqui também para que tenhamos um bom banco de dados sobre nossa área.

Um bom artigo de Osmar Alves de queiroz Filho

http://www.sebraepb.com.br:8080/bte/download/Finan%C3%A7as/58_1_arquivo_quantocobrar.pdf

Site com boas ferramentas que auxiliam no cálculo do preço de serviços de informática e outros

http://edcom.wordpress.com/2006/09/12/quanto-cobrar-pelo-seu-trabalho/

Tarifador de Prestação de Serviços

Programa desenvolvido por Marcelo Souza Lima para ajudar no cálculo do preço de serviços de desenvolvimento ou outros serviços.

Faça o download (tarifador.zip) do site MegaMiner.com:

http://www.megaminer.com/index.php?c=all&t=all&q=tarifador

Abaixo E-Book free sobre Empreendedorismo

Professor Dailton Felipine, do site http://e-commerce.org.br (Comércio eletrônico).

Técnicas de negociação e fundamentos da definição de preços de design, desenvolvimento, web e freelances

http://www.efetividade.net/2007/01/03/tecnicas-de-negociacao-e-definicao-de-precos-de-design-desenvolvimento-web-e-freelances/

Retirado de
"http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Oportunidade_de_Trabalho/Pre%C3%A7os_de_Servi%C3%A7os"
Page categories: PHP

Aplicativos em PHP/Apêndices/Redes

De Wikibooks

< Aplicativos em PHP | Apêndices

16.8 - Redes

```
Rede            IPs             Máscaras

Classe A        10.0.0.0        255.0.0.0

Classe B        172.16.0.0 até
                172.31.255.255  255.255.0.0

Classe C        192.168.0.0 até
                192.168.255.255 255.255.255.0
```

Gateway - número do IP do servidor, utilizado nas máquinas clientes.

Mais detalhes no site - http://www.infowester.com/internetprotocol.php

Descobrir o MAC address de um IP com arping

Autor: Rodrigo

Data: 28/06/2007

Descobrir o MAC address de um IP com arping

Para descobrir o MAC address de um IP:

1. arping -c2 192.168.0.1

Onde 192.168.0.1 é o IP desejado neste exemplo.

Site original - http://www.vivaolinux.com.br/dicas/verDica.php?codigo=8804

Retirado de "http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/Ap%C3%AAndices/Redes"
Page categories: PHP

Aplicativos em PHP/Apêndices/Editores Gráficos

De Wikibooks

< Aplicativos em PHP | Apêndices

Conteúdo

- 1 Gimp
  - 1.1 Vídeo Curso de Gimp em 10 Lições para Iniciantes no site do Antônio Cláudio
  - 1.2 http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=2755
  - 1.3 Edição pesada de imagens com o Gimp (parte 1)
  - 1.4 Edição pesada de imagens com o Gimp (parte 2)
  - 1.5 Edição pesada de imagens com o Gimp (parte 3)
  - 1.6 "Tatuando" pessoas com o Gimp
  - 1.7 Forum O Gimp
  - 1.8 Alguém aí falou em deformação?
  - 1.9 Deformar ou recriar?
- 2 KolourPaint
- 3 InkScape
- 4 Krita
- 5 Xara Xtreme

Gimp

O Gimp é o editor Gráfico com mais recursos do Linux. Alguns o consideram similar ao Adobe Photoshop.

A maioria das distribuições Linux já vem com ele instalado.

Vídeo Curso de Gimp em 10 Lições para Iniciantes no site do Antônio Cláudio

http://www.tuxresources.org/blog/?p=183

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=2755

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=2755

Edição pesada de imagens com o Gimp (parte 1)

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=2755

Edição pesada de imagens com o Gimp (parte 2)

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=3902

Edição pesada de imagens com o Gimp (parte 3)

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=3904

"Tatuando" pessoas com o Gimp

http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=3920

Forum O Gimp

http://www.ogimp.com.br/modules/wfsection/article.php?articleid=31

Alguém aí falou em deformação?

http://vivaolinux.com.br/artigos/impressora.php?codigo=4457

Deformar ou recriar?

http://www.vivaolinux.com.br/artigos/impressora.php?codigo=6621

KolourPaint

Um bom editor gráfico para Linux cujo nome e funções lembra o Pain do Windows mas conta ainda com mais recursos que o Paint.

No Linux Ubuntu ele não vem instalado por padrão mas através do Adicionar/Remover pode ser instalado. Em Exibir marque Todos os aplicativos disponíveis.

É um editor simples não se comparando ao Gimp, mas ideal para pequenas correções e edições de imagens.

InkScape

Outro bom editor de imagens sendo este vetorial e trabalha com arquivos do tipo SVG.

Krita

Outro editor de imagens for Linux com bons recursos.

Xara Xtreme

Site oficial - http://www.xaralx.org/

Recursos (do site):

```
   * Is very fast, very slick to use.
   * Offers some of the most powerful graphics tools available.
   * Is simple to use and learn.
   * Has a clean, un-cluttered user interface. Few floating dialogs, palettes,
menus etc.
   * Is able to create a huge range of outstanding graphics.
   * Has a huge resource of learning material, tutorials, movies, tips and a
very active enthusiastic and growing user community.
```

PHP

versão de Criação 31-08-2021

Limpar cache com php

Você pode excluir o cache do navegador definindo esses cabeçalhos:

```
<?php
header("Expires: Tue, 01 Jan 2000 00:00:00 GMT");
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");
header("Cache-Control: no-store, no-cache, must-revalidate, max-age=0");
header("Cache-Control: post-check=0, pre-check=0", false);
header("Pragma: no-cache");
?>
```

Mostrando erros no PHP

```
ini_set('display_errors', 1);
error_reporting(E_ALL);
```

Gerando Uuid Unicos

```
uniqid(rand(), true)
```

Se é sobre um imagem ou página

```
<?$time = date("H:i:s");?>
<img src="myfile.jpg?time=<?$time;?>">
```

Você pode usar $ time em partes quando não quiser armazenar em cache. Por isso, sempre puxará uma nova imagem. O Versionate é uma abordagem melhor, caso contrário, pode sobrecarregar o servidor. Lembre-se, o cache do navegador não é apenas bom para a experiência do usuário, mas também para o seu servidor.

A função date

Para exibirmos a data atual no formato brasileiro, por exemplo, ficaria assim:

```
<?php
$hoje = date('d/m/Y');
echo $hoje;
```

Com esta mesma função podemos também pegar o horário atual, além da data. Continua bem fácil:

```
<?php
$agora = date('d/m/Y H:i');
echo $agora;
```

O código que tínhamos antes não muda muito. Veja como fica utilizando a classe que acabamos de conhecer:

```
<?php
$agora = new DateTime(); // Pega o momento atual
echo $agora->format('d/m/Y H:i'); // Exibe no formato desejado
```

Caso a gente não passe nenhum parâmetro para o construtor de DateTime, o momento atual será recuperado. Podemos ainda passar uma string informando o momento que queremos, por exemplo:

```
<?php
$agora = new DateTime('now');
$ontem = new DateTime('yesterday');
$dia15 = new DateTime('2019-03-15');
```

Se já tivermos uma string em outro formato, e quisermos criar um objeto do tipo DateTime, podemos utilizar o método createFromFormat, da seguinte forma:

```
<?php
$formato = 'd/m/Y';
$stringDataDia15 = '15/03/2019';
$dia15 = DateTime::createFromFormat($formato, $stringDataDia15);
```

Agora sim, intervalo entre duas datas

```
<?php
$entrada = new DateTime('09:00');
$saida = new DateTime('18:00');
$intervalo = $saida->diff($entrada);
print_r($intervalo);
```

fuso horário

```
<?php
$timezone = new DateTimeZone('America/Sao_Paulo');
$agora = new DateTime('now', $timezone);
```

Repare que nós colocamos o timezone como America/Sao_Paulo. Mas e se quisermos saber outras possibilidades? É possível ver a lista de fusos horários suportados pelo DateTimeZone com o método DateTimeZone::listIdentifiers():

A classe DateTimeImmutable! Ela pode ser utilizada da mesma forma que a classe DateTime, mas as operações que ela expõe retornam uma nova data, e não alteram a original. Pareceu confuso? Vamos ver na prática essa diferença e de quebra aprender a realizar mais operações com as datas:

```
<?php

$umDia = new DateInterval('P1D'); // Intervalo de 1 dia
// Com DateTime:
$hoje = new DateTime();
echo $hoje->format('d/m'); // Saída 20/03
$hoje->add($umDia); // Altera o valor de $hoje
echo $hoje->format('d/m'); // Saída 21/03

// Com DateTimeImmutable
$hoje = new DateTimeImmutable();
echo $hoje->format('d/m'); // Saída 20/03
$amanha = $hoje->add($umDia); // Não altera o valor de $hoje
echo $hoje->format('d/m'); // Saída 20/03
echo $amanha->format('d/m'); // Saída 21/03
```

Diferencias entre datas

Olá pessoal,

Já vimos aqui no blog Como Comparar Data com PHP, então surge uma pequena dúvida: *Como Calcular a quantidade de Dias entre duas Data?*. É exatamente disso que tratar neste post.

No exemplo abaixo vamos usar o comando **strtotime** que gera o timestamp da data no formato textual para que possamos trabalhar com a mesma. Na lógica isso nos permitirá calcular o número de dias considerando a diferença em segundos de cada data.

Veja o exemplo:
code begin:
****

```
<?php
   $data_inicial = '2013-08-01';
   $data_final = '2013-08-16';

   // Calcula a diferença em segundos entre as datas
   $diferenca = strtotime($data_final) - strtotime($data_inicial);
```

```
    //Calcula a diferença em dias
    $dias = floor($diferenca / (60 * 60 * 24));

    echo "A diferença é de $dias entre as datas";
?>
```

```
<?php

$dateStart = new \DateTime('2017-02-08 08:00');
$dateNow   = new \DateTime(date('Y-m-d H:i'));

$dateDiff = $dateStart->diff($dateNow);
```

```
<?php

// códigos do nosso exemplo acima

$result = $dateDiff->h . ' horas e ' . $dateDiff->i . ' minutos';
```

****

code end:

Explicando o script:

O script já esta bem auto-explicativo, porém vale ressaltar o calculo matemático que foi realizado na linha 9.

Após obter a diferença em segundos entre as duas datas (linha 6) basta fazer a divisão deste valor pelo número de segundos de um dia (60*60*24) que ele irá retornar a quantidade de dias (o comando floor foi utilizado para arredondar o número).

Leitura em arquivos

Tal função aceita os seguintes parâmetros:

resource fopen (string $filename , string $mode [, bool $use_include_path [, resource $context]])

Onde:

- **$filename** é o caminho do arquivo que você está tentando abrir. Se você utilizar uma URL, com HTTP ou FTP, a opção {php}allow_url_fopen{/php} deverá estar ativa no **php.ini**. Caso contrário, estará abrindo um arquivo interno do servidor;
- **$mode** é o modo que o PHP vai utilizar para abrir o arquivo (somente leitura, leitura e escrita, e assim por diante). Vou publicar uma tabela logo abaixo;
- **$use_include_path** indica que o arquivo deve ser procurado nos diretórios especificados em {php}include_path{/php} do **php.ini**;
- **$context** – Veja aqui.

code begin:

****

```
<?php
// Meu arquivo
$arquivo = 'meu_arquivo.txt';

// Cria o recurso (abrir o arquivo)
$handle = fopen( $arquivo, 'r' );
?>
```

****

code end:

Perceba que **fopen** sozinha não faz nada, é necessário utilizar mais funções, que serão detalhadas mais adiante neste artigo.

Modos para fopen

Os modos disponíveis dizem ao PHP como manipular o arquivo, veja:

modo	Descrição
'r'	Abre somente para leitura; coloca o ponteiro no começo do arquivo.
'r+'	Abre para leitura e escrita; coloca o ponteiro no começo do arquivo.
'w'	Abre somente para escrita; coloca o ponteiro no começo do arquivo e apaga tudo o que estiver nele. Se o arquivo não existir, tenta criá-lo.
'w+'	Abre para leitura e escrita; coloca o ponteiro no começo do arquivo e apaga tudo o que estiver nele. Se o arquivo não existir, tenta criá-lo.
'a'	Abre somente para escrita; coloca o ponteiro no final do arquivo. Se o arquivo não existir, tenta criá-lo.
'a+'	Abre para leitura e escrita; coloca o ponteiro no final do arquivo. Se o arquivo não existir, tenta criá-lo.
'x'	Cria e abre o arquivo somente para escrita; coloca o ponteiro no começo do arquivo. Se o arquivo já existir, a chamada a **fopen()** falhará, retornando FALSE e gerando um erro de nível E_WARNING. Se o arquivo não existir, tenta criá-lo. Isto é equivalente a especificar as flags *O_EXCL|O_CREAT* para a chamada de sistema *open(2)*.
'x+'	Cria e abre o arquivo para leitura e escrita; coloca o ponteiro no começo do arquivo. Se o arquivo já existir, a chamada a **fopen()** falhará, retornando FALSE e gerando um erro de nível E_WARNING. Se o arquivo não existir, tenta criá-lo. Isto é equivalente a especificar as flags *O_EXCL|O_CREAT* para a chamada de sistema *open(2)*.

Lendo arquivos com PHP

Depois que você abre um arquivo em PHP, é possível realizar duas ações, ler ou escrever (dependendo do modo utilizado para abrir tal arquivo). Uma das funções do PHP que servem para a leitura dos dados que estão no arquivo, é a fread.

code begin:

****

```
<?php
// Meu arquivo
$arquivo = 'meu_arquivo.txt';

// Cria o recurso (abrir o arquivo)
$handle = fopen( $arquivo, 'r' );

// Lê o arquivo (se existir)
$ler = fread( $handle, filesize($arquivo) );

// Mostra dados na tela
echo $ler;

// Fecha o arquivo
fclose($handle);
?>
```

****

code end:

Escrever em arquivos utilizando PHP

Para escrever, utilize a função fwrite.

Tal função depende da função **fopen** (descrita anteriormente), veja:

code begin:

****

```
<?php
// Meu arquivo
$arquivo = 'meu_arquivo.txt';

// Dados para escrever no arquivo
$html = '<p>Eu amo PHP.</p>';

// Cria o recurso (abrir o arquivo)
$handle = fopen( $arquivo, 'a+' );
```

```
// Escreve
$ler = fwrite( $handle, $html );

// Fecha o arquivo
fclose($handle);

// Mostra os dados do arquivo
echo file_get_contents( $arquivo );
?>
```

****

code end:

Apagando um arquivo com PHP

Para apagar, simplesmente utilize unlink com o caminho do arquivo, veja:

code begin:

****

```
// Apaga o arquivo
unlink( $arquivo );
```

****

code end:

Uploads de Arquivos

****

code end:

```
function uuid(){
    return uniqid(rand(), true);
}
function uuidMd5(){
    return time().uniqid(md5(rand()));
}
function uploadSave($file){
        $response=[];
        $uploaddir = 'uploads/';
        $extensao = pathinfo($file['name']);
        $extensao = ".".$extensao['extension'];
        $imagem = time().uniqid(md5(rand())).$extensao;
        $uploadfile = $uploaddir . $imagem;
        if (move_uploaded_file($file['tmp_name'], $uploadfile)) {
            $response=array(
                "status" => 200,
                "msn"    =>"Arquivo Valido e Enviado com Sucesso.",
                "uri"   => $uploadfile
            );
        } else {
            $response=array(
                "status" => 505,
                "msn"    =>"Possível ataque de upload de arquivo!.",
                "uri"   => $uploadfile
            );
        }
        return $response;
}
```

****

code end:

Croos API Php Nativo

****

code end:

```
<?php
//header("Access-Control-Allow-Origin: *");


/**
 *  An example CORS-compliant method.  It will allow any GET, POST, or OPTIONS requests from any
 *  origin.
 *
 *  In a production environment, you probably want to be more restrictive, but this gives you
 *  the general idea of what is involved.  For the nitty-gritty low-down, read:
 *
 *  - https://developer.mozilla.org/en/HTTP_access_control
 *  - https://fetch.spec.whatwg.org/#http-cors-protocol
 *
 */
function cors() {

    // Allow from any origin
    if (isset($_SERVER['HTTP_ORIGIN'])) {
        // Decide if the origin in $_SERVER['HTTP_ORIGIN'] is one
        // you want to allow, and if so:
        header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
        header('Access-Control-Allow-Credentials: true');
        header('Access-Control-Max-Age: 86400');    // cache for 1 day
    }

    // Access-Control headers are received during OPTIONS requests
    if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
```

```
        if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
            // may also be using PUT, PATCH, HEAD etc
            header("Access-Control-Allow-Methods: GET, POST, OPTIONS");

        if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
            header("Access-Control-Allow-Headers:
{$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");

        exit(0);
    }

}
cors();

?>
```

****

code end:

PHP com MySQL

Enviada por:
José Henrique Monteiro De Almeida

Sumário

1. INTRODUÇÃO AO PHP **4**
1.1. O QUE É PHP? 4
1.2. COMO SURGIU A LINGUAGEM PHP? 4
1.3. CARACTERÍSTICAS DA LINGUAGEM PHP 5
1.4. CONFIGURAÇÃO DO IIS 5
1.5. TESTANDO O PHP 5
2. SINTAXE DO PHP **6**
2.1. DELIMITANDO O CÓDIGO PHP 6
2.2. SEPARADOR DE INSTRUÇÕES 6
2.3. COMENTÁRIOS 6
2.4. IMPRIMINDO CÓDIGO HTML 7
3 – VARIÁVEIS **8**
3.1. NOMES DE VARIÁVEIS 8
3.2. TIPOS SUPORTADOS 8
3.3. TRANSFORMAÇÃO DE TIPOS 11
4. OPERADORES **14**
4.1. OPERADORES ARITMÉTICOS 14
4.2. OPERADORES DE STRINGS 14
4.3. OPERADORES DE ATRIBUIÇÃO 14
4.4. OPERADORES BIT A BIT 15
4.5. OPERADORES LÓGICOS 15
4.6. OPERADORES DE COMPARAÇÃO 15
4.7. OPERADORES DE EXPRESSÃO CONDICIONAL 15
4.8. OPERADORES DE INCREMENTO E DECREMENTO 15
5. ESTRUTURAS DE CONTROLE **17**
5.1. BLOCOS 17
5.2. ESTRUTURA IF / ELSE / ELSEIF 17
5.3. ESTRUTURA SWITCH CASE 20
5.4. ESTRUTURAS DE REPETIÇÃO 22
5.4.1. WHILE 22
5.4.2. DO ... WHILE 23
5.4.3. FOR 23
5.5. QUEBRA DE FLUXO 24
5.5.1. BREAK 24
5.5.2. CONTINUE 24
6. FUNÇÕES **26**
6.1. ESCOPO DE VARIÁVEIS **28**
6.2. GLOBAIS 28
6.3. LOCAIS 29
6.4. ESTÁTICAS 29
6.5. CONSTANTES 30
7. UTILIZANDO FORMULÁRIOS HTML **31**
7.1. DEFININDO UM FORMULÁRIO 31
7.2. INTERAGINDO O PHP COM OS FORMULÁRIOS HTML 35
7.3. VARIÁVEL $_POST 35

8. TRABALHANDO COM MYSQL 40

8.1. O BANCO DE DADOS MYSQL 40
8.2. CARACTERÍSTICAS DO MYSQL 41
8.3. O QUE O MYSQL FAZ DE MELHOR 41
8.3. O MYSQL É GRATUITO ? 41
8.4. TIPOS DE DADOS DO MYSQL 42
8.5. COMANDOS BÁSICOS DO SQL 45
8.6. MANIPULANDO DADOS DAS TABELAS 46

9. CONEXÃO COM O MYSQL VIA PHP 49

9.1. ABRINDO E FECHANDO CONEXÃO COM O MYSQL 49
9.2. TRATAMENTO DE RESULTADOS DE QUERY SELECT 52

10. UTILIZANDO HEADERS 55

11. UTILIZANDO COOKIES 56

11.1. O QUE SÃO COOKIES 56
11.2. GRAVANDO COOKIES 56
11.3. LENDO COOKIES GRAVADOS 57

13. TRABALHANDO COM SESSÕES (SESSIONS) 59

13.1. FUNÇÃO SESSION_REGISTER() 59

14. FUNÇÕES DO PHP 60

14.1. FUNÇÕES DIVERSAS 60
14.2. FUNÇÕES DE MANIPULAÇÃO DE ARQUIVOS 66

1. Introdução ao PHP

1.1. O que é PHP?

O PHP é uma linguagem que permite criar sites WEB dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e links. A diferença de PHP com relação a linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas html puro.

Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. Isso pode ser útil quando o programa está lidando com senhas ou qualquer tipo de informação confidencial.

O que diferencia PHP de um script CGI escrito em C ou Perl é que o código PHP fica embutido no próprio HTML, enquanto no outro caso é necessário que o script CGI gere todo o código HTML, ou leia de um outro arquivo.

1.2. Como surgiu a linguagem PHP?

A linguagem PHP foi concebida durante o outono de 1994 por Rasmus Lerdorf. As primeiras versões não foram disponibilizadas, tendo sido utilizadas em sua home-page apenas para que ele pudesse ter informações sobre as visitas que estavam sendo feitas.

A primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como “Personal Home Page Tools” (ferramentas para página pessoal). Era composta por um sistema bastante simples que interpretava algumas macros e alguns utilitários que rodavam “por trás” das home-pages: um livro de visitas, um contador e algumas outras coisas.

Em meados de 1995 o interpretador foi reescrito, e ganhou o nome de PHP/FI, o “FI” veio de um outro pacote escrito por Rasmus que interpretava dados de formulários HTML (Form Interpreter).

Ele combinou os scripts do pacote Personal Home Page Tools com o FI e adicionou suporte a mSQL, nascendo assim o PHP/FI, que cresceu bastante, e as pessoas passaram a contribuir com o projeto.

Estima-se que em 1996, PHP/FI estava sendo usado por cerca de 15.000 sites pelo mundo, e em meados de 1997 esse número subiu para mais de 50.000. Nessa época houve uma mudança no desenvolvimento do PHP.

Ele deixou de ser um projeto de Rasmus com contribuições de outras pessoas para ter uma equipe de desenvolvimento mais organizada. O interpretador foi reescrito por Zeev Suraski e Andi Gutmans. Esse novo interpretador foi à base para a versão 3.

O lançamento do PHP4, ocorrido em 22/05/2000, trouxe muitas novidades aos programadores de PHP. Uma das principais foi o suporte a sessões, bastante útil pra identificar o cliente que solicitou determinada informação.

Além das mudanças referentes a sintaxe e novos recursos de programação, o PHP4 trouxe como novidade um otimizador chamado Zend, que permite a execução muito mais rápida de scripts PHP.A empresa que produz o Zend promete para este ano o lançamento de um compilador de PHP. Códigos compilados serão executados mais rapidamente, além de proteger o fonte da aplicação. Hoje, já se encontra disponível no mercado a versão 5.x do PHP.

1.3. Características da Linguagem PHP

- É uma linguagem de fácil aprendizado;
- Tem suporte a um grande número de bancos de dados como: dBase, Interbase, mSQL, mySQL, Oracle, Sybase, PostgreSQL e vários outros.
- Tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente, HTTP;
- É multiplataforma, tendo suporte aos sistemas Operacionais mais utilizados no mercado;
- Seu código é livre, não é preciso pagar por sua utilização e pode ser alterado pelo usuário na medida da necessidade de cada usuário
- Não precisa ser compilado.

1.4. Configuração do IIS

Antes de começarmos a trabalhar com o PHP, precisamos instalar e configurar o PHP, o MySQL e o IIS (Internet Information Service).

Para configurá-los siga as orientações do tutorial **Instalando e Configurando o PHP, IIS e MySQL** que acompanha o CD do curso de PHP.

1.5. Testando o PHP

Vamos criar um pequeno script em PHP para testarmos e verificarmos se o PHP e o IIS foram instalados e configurados corretamente no seu computador.

Utilizando o Dreamweaver ou o Bloco de Notas, digite o script a seguir.

```
<html>
<head><title>Aprendendo PHP</title></head>
<body>

<?
    echo "Primeiro Exemplo";
?>

</body>
</html>
```

Salve o arquivo como “**primeiro.php**” na pasta **C:\Cursos\alunos\seudiretório**.

Agora, abra o navegador e, na barra de endereços, digite o seguinte endereço:

http://localhost/seunome/primeiro.php, onde seu nome é o diretório virtual que você criou, lembra?

Observe que o seu browser exibiu a frase **Primeiro exemplo**. Sinal de que o PHP e o IIS foram instalados e configurados corretamente no seu computador.

2. Sintaxe do PHP

2.1. Delimitando o código PHP

O código PHP fica embutido no próprio HTML. O interpretador identifica quando um código é PHP pelas seguintes tags:

```
<?php
    comandos;
?>

<script language="php">
    comandos;
</script>

<?
    Comandos;
?>

<%
    comandos;
%>
```

O tipo de tags mais utilizado é o terceiro, que consiste em uma "abreviação" do primeiro. Para utilizá-lo, é necessário habilitar a opção short-tags na configuração do PHP. O último tipo serve para facilitar o uso por programadores acostumados à sintaxe de ASP. Para utilizá-lo também é necessário habilitá-lo no PHP, através do arquivo de configuração ***php.ini***.

2.2. Separador de instruções

Entre cada instrução em PHP é preciso utilizar o **ponto-e-vírgula**, assim como em C, Perl e outras linguagens mais conhecidas. Na última instrução do bloco de script não é necessário o uso do ponto-e-vírgula, mas por questões estéticas recomenda-se o uso sempre.

2.3. Comentários

Há dois tipos de comentários em código PHP:

Comentários de uma linha:

Marca como comentário até o final da linha ou até o final do bloco de código PHP – o que vier antes. Pode ser delimitado pelo caracter "#" ou por duas barras (//). O delimitador "//", normalmente, é o mais utilizado.

Exemplo:

```
<?

    echo "teste"; // este teste é similar ao anterior
    echo "teste"; /# este teste é similar ao anterior
    // sql "teste";

?>
```

Comentários de mais de uma linha:

Tem como delimitadores os caracteres "/*" para o início do bloco e "*/" para o final do comentário. Se o delimitador de final de código PHP (?>) estiver dentro de um comentário, não será reconhecido pelo interpretador.

Exemplo:

```
<?
    echo "teste";

    /*
        este é um comentário com mais de uma linha.

        echo "teste. Este comando echo é ignorado pelo interpretador do PHP por ser um comentário."
    */
?>
```

2.4. Imprimindo código html

Um script php geralmente tem como resultado uma página html, ou algum outro texto. Para gerar esse resultado, deve ser utilizada uma das funções de impressão, echo e print.

Sintaxes:

```
print(argumento);

echo (argumento1, argumento2, ... );

echo argumento;
```

Exemplos:

```
<?

    $a = 10;
    $b = 20;
    $soma = $a + $b;

    //Imprimindo a soma na tela com o comando print.

    print ("A soma é: ". $soma."<p>");

    ////Imprimindo a soma na tela com o comando echo.

    echo ("A soma é: ". $soma."<p>");

    echo "<br>";

    echo $a." ".$b;

?>
```

3 – Variáveis

3.1. Nomes de variáveis

Toda variável em PHP tem seu nome composto pelo caracter **$(dólar)** e uma string, que deve iniciar por uma letra ou o caracter "_". O PHP é **case sensitive**, ou seja, as variáveis $vivas e $VIVAS são diferentes.

Por isso é preciso ter muito cuidado ao definir os nomes das variáveis. É bom evitar os nomes em maiúsculas, pois como veremos mais adiante, o PHP já possui algumas variáveis pré-definidas cujos nomes são formados por letras maiúsculas.

Exemplo de variáveis:

- $teste = nome correto de variável
- teste= nome errado de variável.
- _teste= nome correto de variável

Obs.: Normalmente, a variável é criada utilizando-se, na maioria das vezes, o caracter **$ (dólar)**.

3.2. Tipos Suportados

O PHP suporta os seguintes tipos de dados:

- Inteiro
- Ponto flutuante
- String
- Array
- Objeto

O PHP utiliza checagem de tipos dinâmica, ou seja, uma variável pode conter valores de diferentes tipos em diferentes momentos da execução do script. Por este motivo não é necessário declarar o tipo de uma variável para usá-la. O interpretador PHP decidirá qual o tipo daquela variável, verificando o conteúdo em tempo de execução. Ainda assim, é permitido converter os valores de um tipo para outro desejado, utilizando o ***typecasting*** ou a função ***settype*** (ver adiante).

Inteiros (integer ou long)

Uma variável pode conter um valor inteiro com atribuições que sigam as seguintes sintaxes:

- $vivas = 1234; # inteiro positivo na base decimal
- $vivas = -234; # inteiro negativo na base decimal
- $vivas = 0234; # inteiro na base octal-simbolizado pelo 0 equivale a 156 decimal
- $vivas = 0x34; # inteiro na base hexadecimal(simbolizado pelo 0x) – equivale a 52 decimal.

A diferença entre **inteiros simples** e **long** está no número de bytes utilizados para armazenar a variável. Como a escolha é feita pelo interpretador PHP de maneira transparente para o usuário, podemos afirmar que os tipos são iguais.

Ponto Flutuante (double ou float)

Uma variável pode ter um valor em ponto flutuante com atribuições que sigam as seguintes sintaxes:

- **$vivas = 1.234;**
- **$vivas = 23e4; # equivale a 230.000**

Strings

Strings podem ser atribuídas de duas maneiras:

a) utilizando aspas simples (') – Desta maneira, o valor da variável será exatamente o texto contido entre as aspas (com exceção de \\ e \' – ver exemplo abaixo)

b) utilizando aspas duplas (") – Desta maneira, qualquer variável ou caracter de escape será expandido antes de ser atribuído.

Exemplo:

```
<?
    $teste = "Mauricio";
    $vivas = '---$teste--\n';
    $texto = “Curso de PHP”;
    echo "$vivas";
?>
```

A saída desse script será "---$teste--\n".

```
<?
    $texto = “Curso de PHP”;
    $teste = "Mauricio";
    $vivas = "---$teste---\n";
    echo "$vivas";
?>
```

A saída desse script será "---Mauricio--" (com uma quebra de linha no final).

A tabela seguinte lista os caracteres de escape:

Sintaxe	Significado
\n	Nova linha
\r	Retorno de carro (semelhante a \n)
\t	Tabulação horizontal
\\	A própria barra (\ )
\$	O símbolo $
\’	Aspa simples
\”	Aspa dupla

Arrays

Arrays em PHP podem ser observados como mapeamentos ou como vetores indexados. Mais precisamente, um valor do tipo array é um dicionário onde os índices são as chaves de acesso. Vale ressaltar que os índices podem ser valores de qualquer tipo e não somente inteiros. Inclusive, se os índices forem todos inteiros, estes não precisam formar um intervalo contínuo Como a checagem de tipos em PHP é dinâmica, valores de tipos diferentes podem ser usados como índices de array, assim como os valores mapeados também podem ser de diversos tipos.

Exemplo:

```
<?
    $cor[0] = “amarelo”;
    $cor[1] = “vermelho”;
    $cor[2] = “verde”;
    $cor[3] = “azul”;
    $cor[4] = “anil”;
    $cor[“teste”] = 1;
?>
```

Equivalentemente, pode-se escrever:

```
<?
    $cor = array(0 => “amarelo”, 1 => “vermelho”, 2 => “verde”, 3 => “azul”, 4 => “anil”, teste => 1);
?>
```

Listas

As listas são utilizadas em PHP para realizar atribuições múltiplas.

Através de listas é possível atribuir valores que estão num array para variáveis. Vejamos o exemplo:

Exemplo:

```
list($a, $b, $c) = array(“a”, “b”, “c”);
```

O comando acima atribui valores às três variáveis simultaneamente. É bom notar que só são atribuídos às variáveis da lista os elementos do array que possuem índices inteiros e não negativos. No exemplo acima as três atribuições foram bem sucedidas porque ao inicializar um array sem especificar os índices eles passam a serinteiros, a partir do zero.

Um fator importante é que cada variável da lista possui um índice inteiro e ordinal, iniciando com zero, que serve para determinar qual valor será atribuído. No exemplo anterior temos $a com índice 0, $b com índice 1 e $c com índice 2. Vejamos um outro exemplo:

```
<?
    $arr = array(1=>”um”,3=>”tres”,”a”=>”letraA”,2=>”dois);

    list($a,$b,$c,$d) = $arr;

    echo “$a<br>”;
    echo “$b<br>”;
    echo “$c<br>”;
    echo “$d<br>”;
?>
```

Após a execução do código acima temos os seguintes valores:

$a == null (vazio, não aparece nada)
$b == “um”
$c == “dois”
$d == “tres”

Devemos observar que à variável $a não foi atribuído valor, pois no array não existe elemento com índice 0 (zero).

Outro detalhe importante é que o valor “tres” foi atribuído à variável $d, e não a $b, pois seu índice é 3, o mesmo que $d na lista. Por fim, vemos que o valor “letraA” não foi atribuído a elemento algum da lista pois seu índice não é inteiro.

Os índices da lista servem apenas como referência ao interpretador PHP para realizar as atribuições, não podendo ser acessados de maneira alguma pelo programador.

De maneira diferente do array, uma lista não pode ser atribuída a uma variável, servindo apenas para fazer múltiplas atribuições através de um array.

Booleanos

O PHP não possui um tipo booleano, mas é capaz de avaliar expressões e retornar true ou false, através do tipo integer: é usado o valor 0 (zero) para representar o estado false, e qualquer valor diferente de zero (geralmente 1) para representar o estado true.

3.3. Transformação de tipos

A transformação de tipos em PHP pode ser feita das seguintes maneiras:

a) Coerções

Quando ocorrem determinadas operações (“+”, por exemplo) entre dois valores de tipos diferentes, o PHP converte o valor de um deles automaticamente (coerção). É interessante notar que se o operando for uma variável, seu valor não será alterado.

O tipo para o qual os valores dos operandos serão convertidos é determinado da seguinte forma: Se um dos operandos for float, o outro será convertido para float, senão, se um deles for integer, o outro será convertido para integer.

Exemplo:

```
<?

    //declaração da variável vivas
    //-------------------------------------
    $vivas = “1”; // $vivas é a string “1”
    $vivas = $vivas + 1;// $vivas é o integer 2
    $vivas = $vivas + 3.7;// $vivas é o double 5.7
    $vivas = 1 + 1.5 // $vivas é o double 2.5

    echo “O valor de vivas é: $vivas”;

?>
```

Como podemos notar, o PHP converte string para integer ou double mantendo o valor. O sistema utilizado pelo PHP para converter de strings para números é o seguinte:

- É analisado o início da string. Se contiver um número, ele será avaliado. Senão, o valor será 0 (zero);
- O número pode conter um sinal no início ("+" ou "-");
- Se a string contiver um ponto em sua parte numérica a ser analisada, ele será considerado, e o valor obtido será double;
- Se a string contiver um "e" ou "E" em sua parte numérica a ser analisada, o valor seguinte será considerado como expoente da base 10, e o valor obtido será double;

Exemplos:

```
<?

    $vivas = 1 + "10.5";        // $vivas == 11.5
    $vivas = 1 + "-1.3e3";      // $vivas == -1299
    $vivas = 1 + "teste10.5";           // $vivas == 1
    $vivas = 1 + "10testes";            // $vivas == 11
    $vivas = 1 + " 10testes";           // $vivas == 11
    $vivas = 1 + "+ 10testes";          // $vivas == 1
    echo "O valor de vivas é: $vivas";

?>
```

b) Transformação explícita de tipos

A sintaxe do **typecast** de PHP é semelhante ao C: basta escrever o tipo entre parênteses antes do valor.

Exemplo:

```
<?

    $vivas = 15;                    // $vivas é integer (15)
    $vivas = (double) $vivas        // $vivas é double (15.0)
    $vivas = 3.9                    // $vivas é double (3.9)
    $vivas = (int) $vivas           // $vivas é integer (3)
                                    // o valor decimal é truncado

    echo "O valor de vivas é: $vivas";

?>
```

Os tipos permitidos na ***Transformação explícita*** são:

- (int), (integer) ---------- = muda para integer;
- (real), (double), (float) ------ = muda para float;
- (string)---------- = muda para string;
- (array) ---------- = muda para array;
- (object) ---------- = muda para objeto.

Com a função settype

A função settype converte uma variável para o tipo especificado, que pode ser "integer", "double", "string", "array" ou "object".

Sintaxe:

Settype(nomedavariável,novo tipo da variávbel)

Exemplo:

```
<?

    $vivas = 15;                // $vivas é integer
    settype($vivas,double)      // $vivas é double
    $vivas = 15;                // $vivas é integer
    settype($vivas,string)      // $vivas é string

    echo “O valor de vivas é: $vivas”;

?>
```

4. Operadores

4.1. Operadores Aritméticos

Só podem ser utilizados quando os operandos são números (integer ou float). Se forem de outro tipo, terão seus valores convertidos antes da realização da operação. São eles:

+	adição
-	subtração
*****	multiplicação
/	divisão
%	módulo

4.2. Operadores de Strings

Só há um operador exclusivo para strings:

. concatenação

Exemplo:

```
<?
    $nome = “Gláucio”;
    $sobrenome = “Nascimento”;
    echo $nome.” “.$sobrenome;
?>
```

A saída do script acima será: Gláucio Nascimento

4.3. Operadores de atribuição

Existe um operador básico de atribuição e diversos derivados. Sempre retornam o valor atribuído. No caso dos operadores derivados de atribuição, a operação é feita entre os dois operandos, sendo atribuído o resultado para o primeiro. A atribuição é sempre por valor, e não por referência.

=	atribuição simples
+=	atribuição com adição
-=	atribuição com subtração
*=	atribuição com multiplicação
/=	atribuição com divisão
%=	atribuição com módulo
.=	atribuição com concatenação

Exemplo:

```
<?
    $a = 7;
    $a += 2; // $a passa a conter o valor 9
?>
```

4.4. Operadores bit a bit

Comparam dois números bit a bit.

&	“e” lógico
|	“ou” lógico
^	ou exclusivo
~	não (inversão)
<<	shift left
>>	shift right

4.5. Operadores Lógicos

Utilizados para inteiros representando valores booleanos

and	“e” lógico
or	“ou” lógico
xor	ou exclusivo
!	não (inversão)
&&	“e” lógico
||	“ou” lógico

Existem dois operadores para “e” e para “ou” porque eles têm diferentes posições na ordem de precedência.

4.6. Operadores de Comparação

As comparações são feitas entre os valores contidos nas variáveis, e não as referências. Sempre retornam um valor booleano.

==	igual a
!=	diferente de
<	menor que
>	maior que
<=	menor ou igual a
>=	maior ou igual a

4.7. Operadores de Expressão condicional

Existe um operador de seleção que é **ternário**. Funciona assim:

(expressao1)?(expressao2):(expressao3)

O interpretador PHP avalia a primeira expressão. Se ela for verdadeira, a expressão retorna o valor de expressão2. Senão, retorna o valor de expressão3.

4.8. Operadores de incremento e decremento

- ++ incremento
- decremento

Podem ser utilizados de duas formas: antes ou depois da variável. Quando utilizado antes, retorna o valor da variável antes de incrementá-la ou decrementá-la.

Quando utilizado depois, retorna o valor da variável já incrementado ou decrementado.

Exemplos:

```
<?

    $a = $b = 10;    // $a e $b recebem o valor 10
    $c = $a++;       // $c recebe 10 e $a passa a ter 11
    $d = ++$b;       // $d recebe 11, valor de $b já incrementado

?>
```

5. Estruturas de Controle

As estruturas que veremos a seguir são comuns para as linguagens de programação imperativas, bastando, portanto, descrever a sintaxe de cada uma delas, resumindo o funcionamento.

5.1. Blocos

Um bloco consiste de vários comandos agrupados com o objetivo de relacioná-los com determinado comando ou função. Em comandos como **if, for, while, switch** e em declarações de funções blocos podem ser utilizados para permitir que um comando faça parte do contexto desejado. Blocos em PHP são delimitados pelos caracteres **"{"** e **"}"**. A utilização dos delimitadores de bloco em uma parte qualquer do código não relacionada com os comandos citados ou funções não produzirá efeito algum, e será tratada normalmente pelo interpretador.

Exemplo:

```
<?
    if ($x == $y)
    comando1;
    comando2;
?>
```

Para que comando2 esteja relacionado ao if é preciso utilizar um bloco:

```
<?
    if ($x == $y) {
    comando1;
    comando2;
    }
?>
```

5.2. Estrutura if / else / elseif

O mais trivial dos comandos condicionais é o **if**. Ele testa a condição e executa o comando indicado se o resultado for true (valor diferente de zero). Ele possui duas sintaxes:

Sintaxes:

```
if (expressão)

        comando;


if (expressão) {

        comando;
        . . .
        . . .
        . . .
        comando;
}
```

Para incluir mais de um comando no if da primeira sintaxe, é preciso utilizar um bloco, demarcado por chaves.

Exemplo 1:

```
<?

$a = 10;
$b = 20;

if ($a > $b)

    echo " o Valor de a é: ".$a + $b;

?>
```

Exemplo 2:

```
<?

$a = 10;
$b = 20;

if ($a >= $b) {

    $media = ($a + $b) / 2;
    $resto = $a % $b;//calcula o resto da divisão de A por B
    echo " o Valor de A é: ".$a "<br>";
    echo " o Valor de B é: ".$b <br>;
    echo " o Valor da média é: ".$media <br>;
    echo " O resto da divisão de A por B é: ".$resto;
}

?>
```

Else

O **else** é um complemento opcional para o **if**. Se utilizado, o comando será executado se a expressão retornar o valor **false** (zero). Suas duas sintaxes são:

Sintaxes:

```
if (expressão)
        comando;
else
        comando;
```

ou

```
if (expressão) {
        comando 1;
        comando n
else {
        comando 1;
        comando n
}
```

Exemplos:

```
<?
```

```
        $a = 10;
        $b = 20;

        if ($a > $b) {
                $maior = $a;
        } else {

                $maior = $b;
        }

    ?>
```

O exemplo acima coloca em **$maior** o maior valor entre $a e $b.

Em determinadas situações é necessário fazer mais de um teste, e executar condicionalmente diversos comandos ou blocos de comandos. Para facilitar o entendimento de uma estrutura do tipo:

```
if (expressao1)
        comando1;
else
        if (expressao2)
                comando2;
        else
                if (expressao3)
                        comando3;
                else
                        comando4;
```

foi criado o comando, também opcional **elseif**. Ele tem a mesma função de um **else** e um **if** usados seqüencialmente, como no exemplo acima. Num mesmo if podem ser utilizados diversos **elseif´s**, ficando essa utilização a critério do programador, que deve zelar pela legibilidade de seu script.

Elseif

O comando **elseif** também pode ser utilizado com dois tipos de sintaxe. Em resumo, a sintaxe geral do comando **if** fica das seguintes maneiras:

Sintaxe:

```
if (expressao1)
        comando;
elseif (expressao2)
        comando;
else
        comando;
```

ou

```
if (expressão 1) {
        comando 1;
        comando n;
}
elseif (expressão 2) {
        comando 1;
        comando n;
}
else {
        comando 1;
        comando n;
}
```

Exemplo:

```
<?
```

```
$nota1 = 6;
$nota 2 = 8;
$media = ($nota1 + $nota2) / 2;

if ($media > 7) {
    echo “Média: ”.$media .”<br>”;
    echo “Aluno aprovado.”;
}
elseif ($media <7) {
    echo “Média: ”.$media .”<br>”;
    echo “Aluno reprovado.”;
}
else {
    echo “Média: ”.$media.”<br>”;
    echo “Aluno em recuperação.”;
}

?>
```

5.3. Estrutura Switch Case

O comando **switch** atua de maneira semelhante a uma série de comandos **if** na mesma expressão. Freqüentemente o programador pode querer comparar uma variável com diversos valores, e executar um código diferente a depender de qual valor é igual ao da variável. Quando isso for necessário, deve-se usar o comando **switch**.

Sintaxe:

```
swith ($valor){
    case "____";
        comandos;
        comandos;
        break;
    case "____";
        comandos;
        comandos;
        break;
    case "____";
        comandos;
        comandos;
        break;
    case "____";
        comandos;
        comandos;
        break;
    default;
        comandos;
        comandos;
        break;
```

O exemplo seguinte mostra dois trechos de código que fazem a mesma coisa, sendo que o primeiro utiliza uma série de **if’s** e o segundo utiliza **switch**:

Exemplos 1: Estrutura IF

```
<?
```

```
    if ($i == 0)
        print "i é igual a zero";
    elseif ($i == 1)
        print "i é igual a um";
    elseif ($i == 2)
        print "i é igual a dois";

?>
```

Exemplo 2: Estrutura SWITCH

```
<?
    switch ($i) {
        case 0:
            print "i é igual a zero";
            break;
    case 1:
            print "i é igual a um";
            break;
    case 2:
            print "i é igual a dois";
            break;
    }
?>
```

É importante compreender o funcionamento do **switch** para não cometer enganos. O comando **switch** testa linha a linha os cases encontrados, e a partir do momento que encontra um valor igual ao da variável testada, passa a executar todos os comandos seguintes, mesmo os que fazem parte de outro teste, até o fim do bloco.

Por isso usa-se o comando **break**, quebrando o fluxo e fazendo com que o código seja executado da maneira desejada. Veremos mais sobre o **break** mais adiante.

Exemplo:

```
<?

    switch ($i) {
        case 0:
            print "i é igual a zero";
        case 1:
            print "i é igual a um";
        case 2:
            print "i é igual a dois";
    }

?>
```

No exemplo acima, se **$i** for igual a **zero**, os **três comandos "print"** serão executados. Se **$i** for igual a **1**, os **dois últimos "print"** serão executados. O comando só funcionará da maneira desejada se **$i** for igual a **2**. Em outras linguagens que implementam o comando switch, ou similar, os valores a serem testados só podem ser do tipo inteiro.

Em PHP é permitido usar valores do tipo string como elementos de teste do comando switch. O exemplo abaixo funciona perfeitamente:

Exemplo:

```
<?
    switch ($s) {
        case “casa”:
            print “A casa é amarela”;
        case “arvore”:
            print “A árvore é bonita”;
    }

    ?>
```

5.4. Estruturas de Repetição

As estruturas de repetição são utilizadas quando o programador precisa, por exemplo, repetir o mesmo comando várias vezes. Vamos ver as estruturas de repetição existentes.

5.4.1. While

O **while** é o comando de repetição (laço) mais simples. Ele testa uma condição e executa um comando, ou um bloco de comandos, até que a condição testada seja falsa. Assim como o **if**, o **while** também possui duas sintaxes alternativas:

Sintaxes:

```
While (condição)
        comando;
```

Ou

```
While (condição) {
        comandos;
        comandos;
}
```

Exemplo:

```
<?
    $a = 0;

    while($a i<= 10) {
        echo $a.”<br>";
        $a++;
    }
?>
```

A expressão só é testada a cada vez que o bloco de instruções termina, além do teste inicial. Se o valor da expressão passar a ser **false** no meio do bloco de instruções, a execução segue até o final do bloco. Se no teste inicial a condição for avaliada como **false**, o bloco de comandos não será executado.

O exemplo a seguir mostra o uso do **while** para imprimir os números de 1 a 10:

```
<?

    $i = 1;
    while ($i <=10)
```

```
        print $i++;
        print "<br>";

?>
```

5.4.2. Do ... While

O laço do..while funciona de maneira bastante semelhante ao while, com a simples diferença que a expressão é testada ao final do bloco de comandos.

Sintaxe:

```
do {
    comando
    . . .
    comando
}
while (<condição>);
```

Exemplo:

```
<?
    $a = 10;

    do {
        echo $a++;
        echo "<br>";
    }
    while ($a <=10)
?>
```

5.4.3. For

O tipo de laço mais complexo é o for. Para os que programam em C, C++ ou Java, a assimilação do funcionamento do for é natural. Mas para aqueles que estão acostumados a linguagens como Pascal, há uma grande mudança para o uso do for.

Sintaxe:

```
for (<inicializacao>;<condicao>;<incremento>) {
    <comando>;
    . . .
    <comando>;
}
```

As três expressões que ficam entre parênteses têm as seguintes finalidades:

- **Inicialização:** comando ou seqüência de comandos a serem realizados antes do inicio do laço. Serve para inicializar variáveis.
- **Condição:** Expressão booleana que define se os comandos que estão dentro do laço serão executados ou não. Enquanto a expressão for verdadeira (valor diferente de zero) os comandos serão executados.
- **Incremento:** Comando executado ao final de cada execução do laço.

Um comando for funciona de maneira semelhante a um while escrito da seguinte forma:

```
<inicializacao>
while (<condicao>) {
comandos
...
<incremento>
}
```

Exemplo:

```
<?
    For ($a = 0, $a <10, $a++) {
    Echo " O valor de A é: ". $a;
    Echo "<br>";
    }
?>
```

5.5. Quebra de fluxo

5.5.1. Break

O comando **break** pode ser utilizado em laços de **do, for e while**, além do uso já visto no comando **switch**. Ao encontrar um **break** dentro de um desses laços, o interpretador PHP pára imediatamente a execução do laço, seguindo normalmente o fluxo do script.

Exemplo 1:

```
<?
    $x = 20;
    while ($x < 0) {
        if ($x == 5) {
           echo "Número inválido";
        break;
        }
        echo "Número ".$x."<br>";
        $x--;
    }
?>
```

No trecho de código acima, o laço **while** tem uma condição para seu término normal **($x <= 0)**, mas foi utilizado o **break** para o caso de um término não previsto no início do laço. Assim o interpretador seguirá para o comando seguinte ao laço.

5.5.2. Continue

O comando continue também deve ser utilizado no interior de laços, e funciona de maneira semelhante ao break, com a diferença que o fluxo ao invés de sair do laço volta para o início dele.

Exemplo:

```
<?

    for ($i = 0; $i < 100; $i++) {

        if ($i % 2)
```

```
            continue;
        echo " $i ";
        }

    ?>
```

O exemplo acima é uma maneira ineficiente de imprimir os números pares entre 0 e 99. O que o laço faz é testar se o resto da divisão entre o número e 2 é 0. Se for diferente de zero (valor lógico true) o interpretador encontrará um continue, que faz com que os comandos seguintes do interior do laço sejam ignorados, seguindo para a próxima iteração.

6. Funções

Sintaxe:

```
function nome_da_função([arg1, arg2, arg3]) {
    Comandos;
    ... ;
[return <valor de retorno>];
}
```

Qualquer código PHP válido pode estar contido no interior de uma função. Como a checagem de tipos em PHP é dinâmica, o tipo de retorno não deve ser declarado, sendo necessário que o programador esteja atento para que a função retorne o tipo desejado. É recomendável que esteja tudo bem documentado para facilitar a leitura e compreensão do código. Para efeito de documentação, utiliza-se o seguinte formato de declaração de função:

```
tipo function nome_da_funcao(tipo arg1, tipo arg2, ...);
```

Este formato só deve ser utilizado na documentação do script, pois o PHP não aceita a declaração de tipos. Isso significa que em muitos casos o programador deve estar atento aos tipos dos valores passados como parâmetros, pois se não for passado o tipo esperado não é emitido nenhum alerta pelo interpretador PHP, já que este não testa os tipos.

Valor de retorno

Toda função pode opcionalmente retornar um valor, ou simplesmente executar os comandos e não retornar valor algum.

Não é possível que uma função retorne mais de um valor, mas é permitido fazer com que uma função retorne um valor composto, como listas ou arrays.

Argumentos

É possível passar argumentos para uma função. Eles devem ser declarados logo após o nome da função, entre parênteses, e tornam-se variáveis pertencentes ao escopo local da função. A declaração do tipo de cada argumento também é utilizada apenas para efeito de documentação.

Exemplos:

```
<?
    function imprime($texto) {
        echo $texto;
    }
    imprime("teste de funções");
?>

<?
    function soma($v1, $v2) {
        $soma = $v1+$v2;
        return soma;
    }
?>
```

Passagem de parâmetros por referência

Normalmente, a passagem de parâmetros em PHP é feita por valor, ou seja, se o conteúdo da variável for alterado, essa alteração não afeta a variável original.

Exemplo:

```
<?

    function mais8($numero) {
        $numero += 8;
        echo $numero. "<br>";
    }

    $a = 3;
    mais8($a); //$a continua valendo 3

    echo $a;

?>
```

No exemplo acima, como a passagem de parâmetros é por valor, a função **mais8** é inútil, já que após a execução sair da função o valor anterior da variável é recuperado. Se a passagem de valor fosse feita por referência, a variável **$a** teria **8** como valor.

O que ocorre normalmente é que ao ser chamada uma função, o interpretador salva todo o escopo atual, ou seja, os conteúdos das variáveis. Se uma dessas variáveis for passada como parâmetro, seu conteúdo fica preservado, pois a função irá trabalhar na verdade com uma cópia da variável. Porém, se a passagem de parâmetros for feita por referência, toda alteração que a função realizar no valor passado como parâmetro afetará a variável que o contém.

Há duas maneiras de fazer com que uma função tenha parâmetros passados por referência: indicando isso na declaração da função, o que faz com que a passagem de parâmetros sempre seja assim; e também na própria chamada da função.

Nos dois casos utiliza-se o modificador "&". Vejamos um exemplo que ilustra os dois casos:

Exemplo:

```
<?

    function mais5(&$num1, $num2) {
        $num1 += 5;
        $num2 += 5;
    }

    $a = $b = 1;
    mais5($a, $b);

    /* Neste caso, só $num1 terá seu valor alterado, pois a passagem por referência está definida na
    declaração da função. */

    mais5($a, &$b);

    /* Aqui as duas variáveis terão seus valores alterados. */

?>
```

Argumentos com valores pré-definidos (default)

Em PHP é possível ter valores default para argumentos de funções, ou seja, valores que serão assumidos em caso de nada ser passado no lugar do argumento. Quando algum parâmetro é declarado desta maneira, a passagem do mesmo na chamada da função torna-se opcional.

Exemplo:

```
<?

    function teste($vivas = "testando") {
        echo $vivas;
    }

    teste(); // imprime "testando"
    teste("outro teste"); // imprime "outro teste"

?>
```

É bom lembrar que quando a função tem mais de um parâmetro, o que tem valor default deve ser declarado por último.

Exemplo:

```
<?
    function teste($figura = circulo, $cor) {
        echo "a figura é um ", $figura, " de cor " $cor;
    }

    teste(azul);
?>
```

Contexto

O contexto é o conjunto de variáveis e seus respectivos valores num determinado ponto do programa. Na chamada de uma função, ao iniciar a execução do bloco que contém a implementação da mesma é criado um novo contexto, contendo as variáveis declaradas dentro do bloco, ou seja, todas as variáveis utilizadas dentro daquele bloco serão eliminadas ao término da execução da função.

6.1. Escopo de Variáveis

O escopo de uma variável em PHP define a porção do programa onde ela pode ser utilizada. Na maioria dos casos todas as variáveis têm escopo global.

Entretanto, em funções definidas pelo usuário um escopo local é criado. Uma variável de escopo global não pode ser utilizada no interior de uma função sem que haja uma declaração. Os escopos de variáveis existentes no PHP são:

- Globais;
- Locais;
- Estáticas;
- Constantes;

6.2. Globais

As variáveis globais são, por definição, as variáveis que podem ser acessadas dentro de todo o script. Porém, quando se cria escopo local como nas funções, precisaremos utilizar um tipo de chamada especial, ou seja, definir aquela variável como uma variável global.

Exemplo:

```
<?
    $curso = “PHP”;
    function mostra() {
        global $curso;
        echo $curso;
    }
    mostra();
?>
```

Uma declaração “**global**” pode conter várias variáveis, separadas por vírgulas. Uma outra maneira de acessar variáveis de escopo global dentro de uma função é utilizando um array pré-definido pelo PHP cujo nome é **$GLOBALS**. O índice para a variável referida é o próprio nome da variável, sem o caracter $. O exemplo acima e o abaixo produzem o mesmo resultado:

Exemplo:

```
<?
    $curso = “PHP”;
    function mostra() {
        echo $GLOBAL[“curso”];
        echo $curso;
    }
    mostra();
?>
```

6.3. Locais

As variáveis Locais são os tipos mais restritos dentro do PHP. Elas funcionam apenas dentro desse escopo.

Exemplo:

```
<?
    $curso = “PHP”;
    function mostra() {
        $var_local = “variable local”;
        echo $var_local;
    }
    echo “<b>$var_local</b>”;
?>
```

6.4. Estáticas

As variáveis Estáticas são variáveis que possuem o mesmo tempo de vida das variáveis globais, com a diferença de funcionarem apenas em escopos locais e serem inicializadas apenas uma vez.

Exemplo:

```
<?
    function contador() {
        static $a = 0;
        echo $a++.”<br>”;
    }
    for ($a = 0, $a <=5, $a++) {
        contador();
    }
?>
```

6.5. Constantes

Variáveis Constantes são variáveis que recebem um único valor que não será alterado dentro da função, permanecendo sempre o mesmo valor. Uma constante só pode conter valores escalares, não podendo conter nem um array nem um objeto.

Sintaxe:

int define(string nome_da_constante, mixed valor); onde:

- **string nome_da_contante** é o nome da variável;
- **mixed valor** é o valor que essa constante irá receber.

A função retorna true se for bem-sucedida. Veja um exemplo de sua utilização a seguir:

Exemplo:

```
define ("pi", 3.1415926536);
$circunf = 2*pi*$raio;
```

7. Utilizando formulários HTML

7.1. Definindo um formulário

Por ser uma linguagem de marcação, a sintaxe do HTML na maioria dos casos exige uma “tag” de início e uma de final daquele bloco. É exatamente isso que ocorre com a definição de um formulário: uma tag no início e outra no final, sendo que todos os elementos do formulário devem estar entre as duas tags. Isto torna possível a inclusão de mais de um formulário num mesmo html. As tags citadas são:

<form name=”” action=”” method=”” enctype=””>

Onde temos:

- **name:** o identificador do formulário. Utilizado principalmente em Scripts client-side (JavaScript);
- **action:** nome do script que receberá os dados do formulário ao ser submetido;
- **method:** método de envio dos dados: get ou post;
- **enctype:** formato em que os dados serão enviados. O default é urlencoded. Se for utilizado um elemento do tipo upload de arquivo (file) é preciso utilizar o tipo multipart/form-data.

Exemplo:

<**form action**="exemplo.php" **method**="post">

(textos e elementos do form)

</form>

Cada elemento do formulário deve possuir um nome que irá identificá-lo no momento em que o script indicado no ACTION for tratar os dados.

A tag <input>

Muitos elementos de um formulário html são definidos pela tag <input>. Cada tipo de elemento possui parâmetros próprios, mas todos possuem pelo menos dois parâmetros em comum: **type**, que define o tipo de elemento, e **name**, que como já foi dito define o nome daquele elemento. Existem, ainda, outros parâmetros, também chamados atributos que são comuns a maioria dos campos de um formulário. São eles:

- **Value** - o valor pré-definido do elemento, que aparecerá quando a página for carregada;
- **Size** - O tamanho do elemento na tela, em caracteres;
- **Maxlength** - O tamanho máximo do texto contido no elemento, em caracteres;

Campo de Texto (Text)

Exibe na tela um campo para entrada de texto com apenas uma linha.

Sintaxe:

<input type="**text**" name="" value="" size="" maxlength="">

Campo de Texto com Máscara (Password)

Tipo de campo semelhante ao anterior, com a diferença que neste caso os dados digitados são substituídos por asteriscos, e por isso são os mais recomendados para campos que devam conter senhas. É importante salientar que nenhuma criptografia é utilizada. Apenas não aparece na tela o que está sendo digitado.

Sintaxe:

<input type="**password**" name="" value="" size="" maxlength="">

Checkbox

Utilizado para campos de múltipla escolha, onde o usuário pode marcar mais de uma opção.

Sintaxe:

<input type="**checkbox**" name="" value="" checked>

Onde:

- **Value** - o valor que será enviado ao servidor quando o formulário for submetido, no caso do campo estar marcado;
- **Checked** - O estado inicial do elemento. Quando presente, o elemento já aparece marcado;

Radio Button

Utilizado para campos de múltipla escolha, onde o usuário pode marcar apenas uma opção. Para agrupar vários elementos deste tipo, fazendo com que eles sejam exclusivos, basta atribuir o mesmo nome a todos do grupo.

Sintaxe:

<input type="**radio**" name="" value="" checked>

onde:

- **Value** - o valor que será enviado ao servidor quando o formulário for submetido, no caso do campo estar marcado;
- **Checked** - O estado inicial do elemento. Quando presente, o elemento já aparece marcado;

Upload de arquivos (file)

Exibe na tela do browser um campo de texto e um botão, que ao clicado abre uma janela para localizar um arquivo no disco. Para utilizar este tipo de componente, o formulário deverá utilizar o método "**POST**" e ter o parâmetro "**enctype**" com o valor "**multipart/form-data**".

Sintaxe:

<**form** name="form1" **enctype="multipart/form-data" method="post"** action="">
<input type="**file**" name="" size="30">
</form>

Onde:

- **Size** - O tamanho do campo de texto exibido.

TextArea

Exibe na tela uma caixa de texto, com o tamanho definido pelos parâmetros “cols” e “rows”.

Sintaxe:

```
<textarea cols="" rows="" name="" wrap="">texto</textarea>
```

Onde:

- **Cols** - número de colunas do campo, em caracteres;
- **Rows** - número de linhas do campo, em caracteres;
- **Wrap** - Maneira como são tratadas as quebras de linha automáticas. O valor soft faz com que o texto “quebre” somente na tela, sendo enviado para o servidor o texto da maneira como foi digitado; O valor “hard” faz com que seja enviado para o servidor da maneira como o texto aparece na tela, com todas as quebras de linhas inseridas automaticamente; o valor “off” faz com que o texto não quebre na tela e nem quando enviado ao servidor.

Obs.: O elemento do tipo textarea não possui o parâmetro “*value*”. O valor pré-definido do campo é o texto que fica entre as tags *<textarea> e </textarea>*.

Select

Campo utilizado para que o usuário faça a seleção a partir de uma lista de opções.

Sintaxe:

```
<select name="" size="" multiple>

    <option value="">texto</option>
    <option value="">texto</option>
    <option value="">texto</option>
    <option value="">texto</option>
    <option value="">texto</option>
    <option value="">texto</option>

</select>
```

Onde:

- **Size** - número de linhas exibidas. Default: 1;
- **Multiple** - parâmetro que, se presente, permite que sejam selecionadas duas ou mais linhas, através das teclas Control ou Shift;
- **Option** - Cada item do tipo “option” acrescenta uma linha ao select;
- **Value** - Valor a ser enviado ao servidor se aquele elemento for selecionado. Default: o texto do item;
- **Texto** - valor a ser exibido para aquele item. Não é definido por um parâmetro, mas pelo texto que fica entre as tags <option> e </option>

Se o parâmetro “size” tiver o valor 1 e não houver o parâmetro “multiple”, exibe na tela uma “combo box”. Caso contrário, exibe na tela uma “select list”.

Hidden

Campo oculto que é utilizado para se passar parâmetros para o servidor. Este campo não é visível para o usuário.

Sintaxe:

<input type="**hidden**" name="" value="">

Onde:

- **Value** - o parâmetro que será passado para o servidor.

Submit Button

Utilizado para enviar os dados do formulário para o script descrito na seção “action” da definição do formulário

Sintaxe:

<input type="**submit**" name="" value="">

Onde:

- **Value** - o texto que aparecerá no corpo do botão.

Reset Button

Utilizado para fazer todos os campos do formulário retornem ao valor original, quando a página foi carregada. Bastante utilizado como botão “limpar”, mas na realidade só limpa os campos se todos eles têm como valor uma string vazia.

Sintaxe:

<input type="**reset**" name="" value="">

Onde:

- **Value** - o texto que aparecerá no corpo do botão.

Button

Utilizado normalmente para ativar funções de scripts client-side (JavaScript, por exemplo). Sem essa utilização, não produz efeito algum.

Sintaxe:

<input type="**button**" name="" value="">

Onde:

- **Value** - o texto que aparecerá no corpo do botão.

No próximo tópico vamos ver como fazer para criar a interação do PHP com os formulários. Vamos ver como manipular dados do formulário com o PHP.

7.2. Interagindo o PHP com os Formulários HTML

O que você deverá observar quando criar seus formulários para manipular dados no PHP:

- ✓ Seu formulário deve conter um botão "**SUBMIT**" para poder enviar as informações;
- ✓ Todos os campos do formulário que serão tratados no script PHP devem conter o parâmetro "**NAME**", caso contrário, os dados não serão passados para o script PHP;

Como as informações do formulário são passadas para esse script PHP e como as informações do formulário enviado são tratadas, dependem de você. Existem 2 métodos como as informações podem ser passadas: **GET** e **POST**. O recomendável sempre, para todos os formulários é usar o método **POST**, onde os dados enviados **não são visíveis** nas URLs, ocultando possíveis importantes informações e permitindo o envio de longas informações. O **GET** é totalmente o contrário disso.

7.3. Variável $_POST

Esta variável é utilizada para receber as variáveis vindas do formulário pelo método post.

Sintaxe:

```
$_POST[campo_do_formulário]
```

onde:

campo_do_formulário – é o campo que foi criado no formulário que se deseja recuperar.

Exemplo:

```
<?
    $_POST[campo1];
?>
```

Vamos ver um outro exemplo utilizando um formulário HTML e um script PHP que recebe os dados do formulário.

```
<form action="script.php" method="post">
    Campo 1: <input type=text name=campo1><br>
    Campo 2: <input type=text name=campo2><br>
    <input type=submit value="OK">
</form>
```

O formulário acima usa o método POST para envio das informações, então em PHP, teremos o seguinte script:

```
<?
    echo "O valor de CAMPO 1 é: " . $_POST["campo1"];
    echo "<br>O valor de CAMPO 2 é: " . $_POST["campo2"];
?>
```

Se o formulário tivesse sido enviado usando o método **GET**, você simplesmente usaria **$_GET** no lugar de **$_POST**.

Vamos criar um pequeno exemplo de como se trabalhar com formulários no PHP. Nossa página irá enviar os dados de um formulário para o servidor e exibir esses mesmos dados numa página de resposta criada em PHP.

O formulário que iremos montar a seguir (página **form1.html**) irá solicitar que você preencha alguns dados. Ao clicar num botão ***submit***, o que você digitou e preencheu no formulário, será enviado ao servidor especificado para que possa ser produzida uma resposta, no nosso caso, ao arquivo **RespForm1.php** . O **PHP** trata esses valores como variáveis, cujo nome é o nome do campo definido no formulário. O exemplo abaixo, mostra também que o código **PHP** pode ser inserido em qualquer parte do código **HTML**.

Digite o código HTML a seguir (**form1.html**)

```
<html>
<head><title>Curso de PHP com MySQL</title></head>
<body>
<form name="form1" method="get" action="RespForm1.php">
<p>Nome:<br>
    <input name="nome" type="text" id="nome" size="30" maxlength="30"></p>
<p>Senha:<br><input name="senha" type="password" id="senha" size="10" maxlength="10"></p>
<p>Sexo:<br>
    <input name="sexo" type="radio" value="Masculino">Masculino
    <input name="sexo" type="radio" value="Feminino">Feminino</p>
<p>Selecione o Turno:<br>

    <input name="turno" type="checkbox" id="turno" value="Manhã">Manhã
    <input name="turno" type="checkbox" id="turno" value="Tarde">Tarde
    <input name="turno" type="checkbox" id="turno" value="Noite">Noite</p>

<p>Cidade:<br>
<select name="cidade" id="cidade">

    <option value="vazio"> </option>
    <option value="Rio">Rio de Janeiro</option>
    <option value="São Paulo">São Paulo</option>
    <option value="Belo Horizonte">Belo Horizonte</option>

</select></p>
<p><input type="submit" value="Enviar"></p>
<input type='hidden' name='btnOK' value='1'>
</form>
</body>
</html>
```

Salve o arquivo na sua pasta com o nome **Form1.html**. Agora, vamos criar o arquivo **RespForm1.php**.

Arquivo RespForm1.php

```
<?

    // Recebe os dados do formulário com a variável $_POST

    $nome = $_POST["nome"];
    $senha = $_POST["senha"];
    $sexo = $_POST["sexo"];
    $turno = $_POST["turno"];
    $cidade = $_POST["cidade"];

    // Exibe os dados na página de resposta: RespForm.php

    echo "Os dados recebidos do formulário são: <p>";
    echo "
            <table width = '400' border='1' cellspacing='0' cellpadding='0'>
        <tr>
                <td width = '100'>Nome:<td>
                <td width = '300'>$nome</td>
        </tr>
        <tr>
                <td width = '100'>Senha:<td>
                <td width = '300'>$senha</td>
        </tr>
        <tr>
                <td width = '100'>Sexo:<td>
                <td width = '300'>$sexo</td>
        </tr>
        <tr>
                <td width = '100'>Turno:<td>
                <td width = '300'>$turno</td>
        </tr>
        <tr>
                <td width = '100'>Cidade:<td>
                <td width = '300'>$cidade</td>
        </tr>
        </table>
    ";

?>

<html>

<head><title>Curso de PHP</title>

</head>

<body>

<p><font face="Arial, Helvetica, sans-serif" size="4">

<a href="form1.html">Clique aqui para voltar ao formul&aacute;rio.</a>

</font>

</body>

</html>
```

Após salvar os arquivos **Form1.html** e **RespForm1.php**, abra o **browser** e, na linha de endereço digite:

http://localhost/seudiretorio(o diretório virtual criado no IIS)/Form1.html.

Você verá apenas um formulário que contem os campos para você preencher, conforme mostra a figura 01 a seguir.

Figura 1

Após preencher o formulário, clique no botão **Enviar** para submeter este formulário ao servidor. Uma nova página, que é a resposta do servidor, deverá aparecer conforme mostra a figura 2 a seguir.

Figura 2

Observações:

Em vez de usar **$_GET** ou **$_POST** você pode escrever a variável com o mesmo nome do campo do formulário (no exemplo, $campo1 e $campo2). Mas, esse uso não é recomendado, pois se a diretiva "register_globals" na configuração do seu PHP estiver desativada, as variáveis com nome dos campos dos formulários, terão um valor vazio.

Uma solução para isso é usar a função **import_request_variables** no começo dos seus scripts que interpretam formulários. Essa função aceita 3 letras como argumentos: P, G e C, referentes a $_POST, $_GET e $_COOKIE respectivamente.

Exemplo:

```
<?

import_request_variables("gP");

?>
```

O que acontece?

Exemplo:

Você possui formulário com os campos "nome", "endereço" e "idade". Assuma que a diretiva "register_globals" do seu PHP esteja desligada, mas, você já havia programado o script usando as variáveis no escopo global, no lugar de $_POST.

Adicionando aquela função no começo do script, as variáveis do seu formulário postado:

$_POST["nome"], $_POST["endereco"] e $_POST["idade"] serão extraídas cada para uma variável diferente: $nome, $endereco e $idade.

8. Trabalhando com MySQL

O MySQL surgiu a partir da necessidade da equipe que criou o SGBD, de utilizar algum mecanismo que permitisse a conexão de tabelas criadas na linguagem SQL para um determinado fim. A princípio, o grupo iria utilizar o mSQL, mas logo perceberam que esta ferramenta não era rápida o suficiente para atender às necessidades do projeto. O jeito foi criar uma solução própria. Nascia o MySQL.

O MySQL foi criado por Michael Widenius na companhia suíça TcX. Por volta de 1979 Michael desenvolveu um banco de dados chamado UNIREG, sendo rescritos em várias linguagens desde então" [YAR 99]. Em 1994, a empresa TcX começou o desenvolvimento de aplicações baseadas na Web, tendo como base o banco UNIREG, porém esse banco possuía muito "overhead" para obter sucesso em uma aplicação para geração de páginas dinâmicas na Web. Então a empresa TcX começou a procurar por outro banco o mSQL, uma ferramenta baseada em SQL mas com características pobres não possuindo por exemplo suporte a índices, e com desempenho inferior ao UNIREG.

Foi então que o desenvolvedor do banco UNIREG contatou o David Hughes criador do mSQL, para saber do interesse dele em unir os dois bancos. Sendo positivo o interesse de David , a empresa TcX resolveu desenvolver um novo banco, mas mantendo ao máximo a compatibilidade com mSQL. TcX foi esperta o suficiente para não reinventar o que já estava bem feito, ela construiu seu servidor baseado na estrutura que já estava montada do UNIREG e utilizou grande número de utilitários escritas para mSQL e fez API's para o novo servidor praticamente iguais ao mSQL. Como resultado usuários do mSQL que decidissem mudar para o novo servidor da TcX, teriam apenas que fazer pequenas e simples mudanças nos códigos existentes.

Então foi em maio de 1995 que, definitivamente, a primeira versão do MySQL foi lançada. Um dos parceiros da TcX sugeriu a distribuição do servidor na Internet, o objetivo disso era a utilização de um modelo pioneiro desenvolvido por Aladdin Peter Deutsch. O resultado foi um maior flexibilidade em sem "copyright", que fez do MySQL mais difundido gratuitamente do que mSQL.

8.1. O Banco de Dados MySQL

O MySQL foi desenvolvido pela TCX em 1996. Atualmente a MySQL AB desenvolve o programa. MySQL AB é a companhia dos fundadores e principais desenvolvedores do MySQL. Eles criaram-no porque precisavam de um banco de dados relacional que pudesse tratar grandes quantidades de dados em máquinas de custo relativamente barato.

É um dos bancos de dados relacionais mais rápidos do mercado, apresenta quase todas as funcionalidades dos grandes bancos de dados . MySQL é uma linguagem simples, em que você facilmente pode gravar, alterar e recuperar informações num web site com segurança e rapidez O MYSQL é executado, principalmente, em sistemas que participam da filosofia UNIX, embora outros sistemas S.O também fornecem suporte, como Windows, por exemplo.

O Servidor MySQL foi desenvolvido originalmente para lidar com bancos de dados muito grandes de maneira muito mais rápida que as soluções existentes e tem sido usado em ambientes de produção de alta demanda por diversos anos de maneira bem sucedida. Apesar de estar em constante desenvolvimento, o Servidor MySQL oferece hoje um rico e proveitoso conjunto de funções. A conectividade, velocidade, e segurança fazem com que o MySQL seja altamente adaptável para acessar bancos de dados na Internet.

O Programa de Banco de Dados MySQL é um sistema cliente/servidor que consiste de um servidor SQL multitarefa que suporta acessos diferentes, diversos programas clientes e bibliotecas, ferramentas administrativas e diversas interfaces de programação (API's). Também concedemos o Servidor MySQL como uma biblioteca multitarefa que você pode ligar à sua aplicação para chegar a um produto mais rápido, menor e mais facilmente gerenciável.

8.2. Características do MySQL

- Multi-plataforma, portanto suporta diferentes plataformas: Win32, Linux, FreeBSD, Unix, etc...
- Suporte a múltiplos processadores
- Um sofisticado sistema de senhas criptografadas flexível e Seguro.
- Suporta até 16 índices por tabela
- Código fonte escrito em C e C++ e testado com uma variedade de diferentes compiladores
- As tabelas criadas podem ter tamanho de até 4 GB
- Banco de dados de código aberto e gratuito
- Suporte as API´s das Seguintes linguagens: PHP, Perl, C,C++,Java, Pynthon, etc...
- Suporte à ODBC, você pode facilmente conectar o Access a um banco de dados do MySQL
- O Cliente conecta no MySQL através de conexões TCP/IP.
- Nenhum problema com o Y2K, visto que o MySQL usa o relógio do Unix que não apresentará problemas até 2069
- Capacidade para manipular bancos com até 50 milhões de registros
- Reduz a administração, engenharia e a sustentação custa por até 50%

8.3. O que o MySQL faz de melhor

- Aplicações Web
- Suporte a código fonte aberto
- Requisitos de sistema baixo
- Tabelas com tamanho grande
- Estabilidade

8.3. O MySQL é gratuito ?

Pessoas confundem "free" com "grátis" o que é comum aqui no Brasil. Mas em se tratando de software este "free" é de open source e não gratuito.

Para poder utilizar o MySQL sob a licença GPL e não precisar pagar, o produto desenvolvido precisa ser GPL também, senão, orientamos a compra da licença comercial, com baixo custo, sendo comercializada por servidor, sem limites de usuários e processadores e ainda com garantia perpétua de atualização de versão para o resto da vida.

8.4. Tipos de dados do MySQL

Os tipos de dados que pode ter um campo, podem-se agrupar em três grandes grupos:

1. Tipos numéricos
2. Tipos de Data
3. Tipos de Cadeia

1) Tipos numéricos:

Existem tipos de dados numéricos, que se podem dividir em dois grandes grupos, os que estão em vírgula flutuante (com decimais) e os que não.

TinyInt: é um número inteiro com ou sem sinal. Com sinal a margem de valores válidos é desde -128 até 127. Sem sinal, a margem de valores é de 0 até 255

SmallInt: número inteiro com ou sem sinal. Com sinal a margem de valores válidos é desde -32768 até 32767. Sem sinal, a margem de valores é de 0 até 65535.

MediumInt: número inteiro com ou sem sinal. Com sinal a margem de valores válidos é desde -8.388.608 até 8.388.607. Sem sinal, a margem de valores é de 0 até 16777215.

Int: número inteiro com ou sem sinal. Com sinal a margem de valores válidos é desde -2147483648 até 2147483647. Sem sinal, a margem de valores é de 0 até 429.496.295

BigInt: número inteiro com ou sem sinal. Com sinal a margem de valores válidos é desde -9.223.372.036.854.775.808 até 9.223.372.036.854.775.807. Sem sinal, a margem de valores é de 0 até 18.446.744.073.709.551.615.

Float: número pequeno em vírgula flutuante de precisão simples. Os valores válidos vão desde -3.402823466E+38 até -1.175494351E-38,0 eté desde 175494351E-38 até 3.402823466E+38.

Double: número em vírgula flutuante de dupla precisão. Os valores permitidos vão desde -1.7976931348623157E+308 até -2.2250738585072014E-308, 0 e desde 2.2250738585072014E-308 até 1.7976931348623157E+308

Decimal: Número em vírgula flutuante desempacotado. O número armazena-se como uma cadeia.

Tipo de Campo	Tamanho de Armazenamento
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
BIGINT	8 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes
DECIMAL(M,D)	M+2 bytes se D > 0, M+1 bytes se D = 0

2) Tipos data:

Na hora de armazenar datas, há que ter em conta que MySQL não verifica de uma maneira estricta se uma data é válida ou não. Simplesmente comprova que o mês está compreendido entre 0 e 12 e que o dia está compreendido entre 0 e 31.

Date: tipo data, armazena uma data. A margem de valores vai desde o 1 de Janeiro de 1001 ao 31 de dezembro de 9999. O formato de armazenamento é de ano-mes-dia.

DateTime: Combinação de data e hora. A margem de valores vai desde o 1 ed Janeiro de 1001 às 0 horas, 0 minutos e 0 segundos ao 31 de Dezembro de 9999 às 23 horas, 59 minutos e 59 segundos. O formato de armazenamento é de ano-mes-dia horas:minutos:segundos

TimeStamp: Combinação de data e hora. A margem vai desde o 1 de Janeiro de 1970 ao ano 2037. O formato de armazenamento depende do tamanho do campo:

Tamanhodo campo	Formato
14	AnoMesDiaHoraMinutoSegundo aaaammddhhmmss
12	AnoMesDiaHoraMinutoSegundo aammddhhmmss
8	AnoMesDia aaaammdd
6	AnoMesDia aammdd
4	AnoMes aamm
2	Ano aa

Time: armazena uma hora. A margem de horas vai desde -838 horas, 59 minutos e 59 segundos. O formato de armazenamento é 'HH:MM:SS'.

Year: armazena um ano. A margem de valores permitidos vai desde o ano 1901 ao ano 2155. O campo pode ter tamanho dois ou tamanho 4 dependendo de se queremos armazenar o ano com dois ou quatro algarismos.

Tipo de Campo	Tamanho de Armazenamento
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte

3 Tipos de cadeia:

Char(n): armazena uma cadeia de longitude fixa. A cadeia poderá conter desde 0 até 255 caracteres.

VarChar(n): armazena uma cadeia de longitude variável. A cadeia poderá conter desde 0 até 255 caracteres. Dentro dos tipos de cadeia pode-se distinguir dois subtipos, os tipo Test e os tipo Blob (Binary Large Object) A diferença entre um tipo e outro é o tratamento que recebem na hora de ordená-los e compará-los. No tipo test ordena-se sem ter importância as maiúsculas e as minúsculas e no tipo blob ordena-se tendo em conta as maiúsculas e minúsculas.

TinyText e TinyBlob: Coluna com uma longitude máxima de 255 caracteres.

Blob e Text: um texto com um máximo de 65535 caracteres.

MediumBlob e MediumText: um texto com um máximo de 16.777.215 caracteres.

LongBlob e LongText: um texto com um máximo de caracteres 4.294.967.295. Há que ter em conta que devido aos protocolos de comunicação os pacotes podem ter um máximo de 16 Mb.

Enum: campo que pode ter um único valor de uma lista que se especifica. O tipo Enum aceita até 65535 valores diferentes.

Tipo de campo	**Tamanho de Armazenamento**
CHAR(n)	n bytes
VARCHAR(n)	n +1 bytes
TINYBLOB, TINYTEXT	Longitude+1 bytes
BLOB, TEXT	Longitude +2 bytes
MEDIUMBLOB, MEDIUMTEXT	Longitude +3 bytes
LONGBLOB, LONGTEXT	Longitude +4 bytes
ENUM('value1','value2',...)	1 ó dos bytes dependendo do número de valores

8.5. Comandos Básicos do SQL

Iremos ver os comandos básicos do SQL, utilizados pela maioria dos bancos de dados , inclusive o MySQL, que serão necessários para o desenvolvimento do nosso projeto, pois o objetivo deste curso não é aprender o SQL, mas sim, o PHP.

Observe que todo comando SQL termina com um **; (ponto e vírgula)**.

Comando Create Database

Este comando permite a criação do banco de dados.

Sintaxe:

```
CREATE DATABASE < nome_db >;
```

onde:

- **nome_db** - indica o nome do Banco de Dados a ser criado.

Exemplo:

```
Create database focus;
```

Comando Create Table

Este comando permite a criação de tabelas no banco de dados.

Sintaxe:

```
CREATE TABLE < nome_tabela > (
nome_atributo1 < tipo > [ NOT NULL ],
nome_atributo2 < tipo > [ NOT NULL ],
......
nome_atributoN < tipo > [ NOT NULL ]
) ;
```

onde:

- **nome_table** - indica o nome da tabela a ser criada.
- **nome_atributo** - indica o nome do campo a ser criado na tabela.
- **tipo -** indica a definição do tipo de atributo (integer(n), char(n), ...).

Exemplo:

```
Create table alunos (
Id_aluno UNSIGNED INT(3) NOT FULL,
nome CHAR(40) NOT NULL,
endereco CHAR (50) NOT FULL
turma CHAR(20) NOT NULL,
PRIMARY KEY (matricula)
);
```

Comando Drop

Este comando elimina a definição da tabela, seus dados e referências.

Sintaxe:

DROP TABLE < nome_tabela > ;

Exemplo:

Drop table alunos;

Comando Alter

Este comando permite inserir/eliminar atributos nas tabelas já existentes.

Sintaxe:

ALTER TABLE < nome_tabela > ADD / DROP (
nome_atributo1 < tipo > [NOT NULL],
nome_atributoN < tipo > [NOT NULL]
) ;

Exemplo:

Alter table alunos **ADD COLUMN** turno char(10) NOT NULL;

8.6. Manipulando dados das tabelas

Comando SELECT

Permite recuperar informações existentes nas tabelas.

Sintaxe:

SELECT [DISTINCT] expressao [AS nom-atributo] [FROM from-list] [WHERE condicao] [ORDER BY attr_name1 [ASC | DESC]

onde:

- **DISTINCT** - Para eliminar linhas duplicadas na saída.
- **Expressão** - Define os dados que queremos na saída, normalmente uma ou mais colunas de uma tabela da lista FROM.
- **AS nom-atributo** - um alias para o nome da coluna, exemplo:
- **FROM** - lista das tabelas na entrada
- **WHERE** - critérios da seleção
- **ORDER BY** - Critério de ordenação das tabelas de saída. Podem ser:
    - **ASC** - ordem ascendente (crescente);
    - **DESC** - ordem descendente (decrescente)

Exemplo:

Select cidade, estado **from** brasil **where** populacao > 100000 **order by** Desc;

Comando INSERT

Adiciona um ou vários registros a uma tabela. Isto é referido como consulta anexação.

Sintaxe:

INSERT INTO destino [(campo1[, campo2[, ...]])]
VALUES (valor1[, valor2[, ...])

Onde;

- **Destino -** O nome da tabela ou consulta em que os registros devem ser anexados.
- **campo1, campo2** - Os nomes dos campos aos quais os dados devem ser anexados
- **valor1, valor2** - Os valores para inserir em campos específicos do novo registro. Cada valor é inserido no campo que corresponde à posição do valor na lista: Valor1 é inserido no campo1 do novo registro, valor2 no campo2 e assim por diante.

Os valores devem ser separados com uma vírgula e os campos de textos entre aspas duplas ou simples.

Exemplo:

Insert into alunos (Id_aluno, nome, endereço, turma, turno)
Values (1, 'Glaucio', 'Av. das Américas', '1101', 'manhã');

Comando UPDATE

Cria uma consulta atualização que altera os valores dos campos em uma tabela especificada com base em critérios específicos.

Sintaxe:

UPDATE tabela SET campo1 = valornovo, ... WHERE critério;

Onde:

- **Tabela** - O nome da tabela cujos dados você quer modificar.
- **Valornovo** - Uma expressão que determina o valor a ser inserido em um campo específico nos registros atualizados.
- **critério** - Uma expressão que determina quais registros devem ser atualizados. Só os registros que satisfazem a expressão são atualizado.

Exemplo:

Update alunos **Set** turno = 'tarde' **where** turma = '1101';

UPDATE é especialmente útil quando você quer alterar muitos registros ou quando os registros que você quer alterar estão em várias tabelas. Você pode alterar vários campos ao mesmo tempo.

UPDATE não gera um conjunto de resultados. Se você quiser saber quais resultados serão alterados, examine primeiro os resultados da consulta seleção que use os mesmos critérios e então execute a consulta atualização.

Comando DELETE

Remove registros de uma ou mais tabelas listadas na cláusula FROM que satisfaz a cláusula WHERE.

Sintaxe:

```
DELETE [tabela.*]
FROM tabela
WHERE critério
```

onde:

tabela.* - O nome opcional da tabela da qual os registros são excluídos.

tabela - O nome da tabela da qual os registros são excluídos.

critério - Uma expressão que determina qual registro deve ser excluído.

Exemplo:

Delete from alunos **WHERE** turno='Manhã';

DELETE é especialmente útil quando você quer excluir muitos registros.

Para eliminar uma tabela inteira do banco de dados, você pode usar o método Execute com uma instrução DROP.

Entretanto, se você eliminar a tabela, a estrutura é perdida. Por outro lado, quando você usa DELETE, apenas os dados são excluídos. A estrutura da tabela e todas as propriedades da tabela, como atributos de campo e índices, permanecem intactos.

Você pode usar DELETE para remover registros de tabelas que estão em uma relação um por vários com outras tabelas. Operações de exclusão em cascata fazem com que os registros das tabelas que estão no lado "vários" da relação sejam excluídos quando os registros correspondentes do lado "um" da relação são excluídos na consulta.

Por exemplo, nas relações entre as tabelas Clientes e Pedidos, a tabela Clientes está do lado "um" e a tabela Pedidos está no lado "vários" da relação. Excluir um registro em Clientes faz com que os registros correspondentes em Pedidos sejam excluídos se a opção de exclusão em cascata for especificada.

O DELETE exclui registros inteiros e não apenas dados em campos específicos. Se você quiser excluir valores de um campo específico, crie uma consulta atualização que mude os valores para Null.

Após remover os registros usando uma consulta exclusão, você não poderá desfazer a operação. Se quiser saber quais arquivos foram excluídos, primeiro examine os resultados de uma consulta seleção que use o mesmo critério e então, execute a consulta exclusão. Mantenha os backups de seus dados. Se você excluir os registros errados, poderá recuperá-los a partir dos seus backups.

9. Conexão com o MySQL via PHP

Todos os exemplos e arquivos do nosso projeto referentes a acesso de banco de dados utilizarão o gerenciador de banco de dados MySQL que encontra-se disponível no CD que acompanha este curso de PHP. Estamos utilizando a versão 5.x do MySQL. No CD estão disponíveis as versões 4.x e 5.x.

Para interagir com uma base de dados SQL existem três comandos básicos que devem ser utilizados: um que faz a conexão com o servidor de banco de dados, um que seleciona a base de dados a ser utilizada e um terceiro que executa uma "query" (consulta) SQL. Existe, ainda, um quarto comando que pode ou ser utilizado, que é o comando que fecha a conexão com o banco de dados.

São eles:

9.1. Abrindo e fechando conexão com o MySQL

Mysql_connect( )

Comando utilizado para criar a conexão com a base de dados num servidor mySQL.

O comando **mysql_pconnect** também pode ser utilizado para criar a conexão. A diferença entre os dois comandos é que o **mysql_pconnect** estabelece uma conexão permanente, ou seja, que não é encerrada ao final da execução do script. Utilizaremos a primeira opção: ***mysql_connect***.

Sintaxe:

mysql_connect(string [host[:porta]] , string [login] , string [senha]) [or die ("mensagem de erro")];

onde:

- **string [host[:porta]]** – é o endereço do servidor, onde o banco está armazenado;
- **string [login]** – é o usuário do banco de dados;
- **string [senha]** – é a senha do usuário do banco de dados MySQL.
- **die** – parâmetro opcional que exibe uma mensagem indicando que a conexão não foi efetuada.

Exemplo:

```
<?
    $conexao = mysql_connect ("localhost", "root", "focus") or die ("Conexão não efetuada");
?>
```

O valor de retorno é um inteiro que identifica a conexão, ou falso se a conexão falhar. Antes de tentar estabelecer uma conexão, o interpretador PHP verifica se já existe uma conexão estabelecida com o mesmo host, o mesmo login e a mesma senha. Se existir, o identificador desta conexão é retornado. Senão, uma nova conexão é criada.

Assim, se a conexão for bem sucedida (existir um servidor no endereço especificado que possua o usuário com a senha fornecida), o identificador da conexão fica armazenado na variável $conexao, caso contrário será mostrada a mensagem "Conexão não efetuada".

Mysql_close( )

Comando utilizado para encerrar uma conexão estabelecida com o comando mysql_connect antes de chegar ao final do script. Caso esse comando não seja utilizado a conexão é encerrada no final do script.

Sintaxe:

mysql_close(int [string da conexão]);

Se o identificador não for fornecido, a última conexão estabelecida será encerrada.

Exemplo:

```
<?
    mysql_close ($conexao);
?>
```

Obs.: O comando mysql_close não encerra conexões estabelecidas com o *mysql_pconnect*.

Mysql_select_db( )

Comando utilizado para selecionar a base de dados depois que a conexão for estabelecida. Se nenhuma de conexão é especificado, a ultima conexão aberta é assumida. Se nenhuma conexão esta aberta, a função irá tentar abrir uma conexão como se **mysql_connect()** fosse chamada sem argumentos e usá-la.

Sintaxe:

mysql_select_db(banco de dados, [string de conexao]) or die [("mensagem")];

Onde:

- **banco de dados – é o banco de dados que será utililzado;**
- **string de conexão – é a conexão criada com o servidor MySQL.**
- **die** – parâmetro opcional que exibe uma mensagem indicando que a conexão não foi efetuada.

Exemplo:

```
<?
    $conexao = mysql_connect("localhost", "root", "focus");

    if ($conexao) {
      die ("Conexão não estabelecida");
    }

    db_select = mysql_select_db("focus", $conexao);
?>
```

Mysql_query( )

Este comando é utilizado para realizar uma consulta SQL no MySQL.

Sintaxe:

Mysql_query(string da consulta);

Onde:

String da consulta – é um dos comandos utilizados do SQL para efetuar uma consulta, uma inclusão, uma alteração ou uma exclusão no banco de dados.

O comando mysql_query() envia uma query (consulta) para o banco de dados ativo no servidor da conexão informada em **string da consulta**. Se o parâmetro **string da consulta** não é especificado, a ultima conexão aberta é usada. Se nenhuma conexão esta aberta, a função tenta estabelecer uma conexão como mysql_connect() seja chamada sem argumentos e usá-la. O resultado é guardado em buffer.

Exemplo:

```
<?

    sql = “select * from alunos where id_aluno = 10”;

    mysql_query (sql);

?>
```

Vamos ver um exemplo de um script PHP utilizando os comandos vistos acima que cria uma conexão com o banco de dados Focus, seleciona a tabela alunos e insere alguns dados na tabela.

```
<?
    $conexao = mysql_connect (“localhost”, “root”, “focus”);
    mysql_select_db(“focus”, $conexão);

    $insere = “Insert into alunos (id_aluno, nome, endereço, turma, turno)
    Values (1, ‘Glaucio’, ‘Av. das Américas’, ‘1101’, ‘manhã’)”;

    $insere1 = “Insert into alunos (id_aluno, nome, endereço, turma, turno)
    Values (2, ‘Alexandre, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

    $insere2 = “Insert into alunos (id_aluno, nome, endereço, turma, turno)
    Values (3, ‘Lucinaldo’, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

    mysql_query ($insere, $conexao) or die (“Não foi possível executar a inserção.”);
    mysql_query ($insere1, $conexao) or die (“Não foi possível executar a inserção.”);
    mysql_query ($insere2, $conexao) or die (“Não foi possível executar a inserção.”);

    $delete = Delete from alunos where turno = “tarde”);

    mysql_query ($delete, $conexao);

    echo (“ todos os alunos do turno da tarde foram excluídos.”);
    mysql_close ($conexao);
?>
```

9.2. Tratamento de resultados de query Select

Ao executar uma query SQL SELECT através do comando mysql_query, o identificador do resultado deve ser armazenado numa variável que pode ser tratada de diversas formas. Duas maneiras interessantes de fazê-lo usam um dos comandos a seguir: **mysql_result, mysql_fetch_row ou mysql_fetch_array**.

Mysql_result( )

Esta função retorna o resultado de uma query SQL.

Sintaxe:

mysql_result(resultado, linha, mixed [campo]);

Onde:

- **resultado** - é o identificador do resultado, obtido com o retorno da função mysql_query;
- **linha** - especifica o registro a ser exibido, já que uma query SELECT pode retornar diversos registros;
- **campo** - é o identificador do campo a ser exibido, sendo o tipo descrito como mixed pela possibilidade de ser de diversos tipos (neste caso, inteiro ou string).

Exemplo:

```
<?
    $conexao = mysql_connect ("localhost", "root", "focus");
    mysql_select_db("focus", $conexão);

    $insere = "Insert into alunos (id_aluno, nome, endereço, turma, turno)
              Values (1, 'Glaucio', 'Av. das Américas', '1101', 'manhã')";

    $insere1 = "Insert into alunos (id_aluno, nome, endereço, turma, turno)
              Values (2, 'Alexandre, 'Av. das Américas', '1101', 'tarde')";

    mysql_query ($insere, $conexao) or die ("Não foi possível executar a inserção.");
    mysql_query ($insere1, $conexao) or die ("Não foi possível executar a inserção.");

    $consulta = "Select Id_Aluno, nome, turno from alunos";
    $resultado = mysql_query ($consulta, $conexao);
    $nome = mysql_result($resultado,0,"nome");
    $turno = mysql_result($resultado,0,"turno");

    echo ("Nome: ".$nome."<p>"."Turno: ".$turno);

    mysql_close ($conexao);
?>
```

Com o exemplo acima, o resultado será:

Nome: Gláucio

Turno: Manhã

Mysql_fetch_array( )

Esta função lê uma linha do resultado e devolve um array, cujos índices são os nomes dos campos. A execução seguinte do mesmo comando lerá a próxima linha, até chegar ao final do resultado.

Sintaxe:

Mysql_fecth_array(string da cosulta SQL);

Exemplo:

```
<?

    $conexao = mysql_connect ("localhost", "root", "focus");
    mysql_select_db("focus", $conexão);

    $insere = "Insert into alunos (id_aluno, nome, endereço, turma, turno)
               Values (1, 'Glaucio', 'Av. das Américas', '1101', 'manhã')";

    $insere1 = "Insert into alunos (id_aluno, nome, endereço, turma, turno)
               Values (2, 'Alexandre, 'Av. das Américas', '1101', 'tarde')";

    mysql_query ($insere, $conexao) or die ("Não foi possível executar a inserção.");
    mysql_query ($insere1, $conexao) or die ("Não foi possível executar a inserção.");

    $consulta = "Select Id_Aluno, nome, turno from alunos";
    $resultado = mysql_query ($consulta, $conexao);

    $dados = mysql_fecth_array($resultado);

    $nome = $dados["nome"];
    $turno =$dados[turno];

    echo ("Nome: ".$nome."<p>"."Turno: ".$turno);

    mysql_close ($conexao);

?>
```

Com o exemplo acima, o resultado será:

Nome: Gláucio
Turno: Manhã

Mysql_fetch_row( )

Esta função é semelhante a função mysql_fetch_array, com a diferença que os índices do array são numéricos, iniciando pelo 0 (zero).

Sintaxe:

Mysql_fetch_row(string de consulta);

Exemplo:

O exemplo anterior substituindo a função **mysql_fecth_array** pela **mysql_fetch_row**.

Mysql_free_result( )

Esta função libera a memória do resultado de uma consulta.

Sintaxe:

mysql_free_result (resource result)

Exemplo:

```
<?

    $conexao = mysql_connect (“localhost”, “root”, “focus”);
    mysql_select_db(“focus”, $conexão);

    $insere = “Insert into alunos (id_aluno, nome, endereço, turma, turno)
              Values (1, ‘Glaucio’, ‘Av. das Américas’, ‘1101’, ‘manhã’)”;

    $insere1 = “Insert into alunos (id_aluno, nome, endereço, turma, turno)
              Values (2, ‘Alexandre, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

    mysql_query ($insere, $conexao) or die (“Não foi possível executar a inserção.”);
    mysql_query ($insere1, $conexao) or die (“Não foi possível executar a inserção.”);

    $consulta = “Select Id_Aluno, nome, turno from alunos”;
    $resultado = mysql_query ($consulta, $conexao);

    $dados = mysql_fecth_array($resultado);

    $nome = $dados[“nome”];
    $turno =$dados[turno];

    echo (“Nome: “.$nome.”<p>”.“Turno: “.$turno);

    mysql_free_result($resultado);

    mysql_close ($conexao);

?>
```

No exemplo acima, a função **mysql_free_result( )** irá liberar toda a memória usada com o identificador de resultado $resultado.

Obs.: mysql_free_result() somente precisa ser chamado se você esta preocupado em quanta memória esta sendo usada para query num grande conjunto de resultados. Toda a memória usada do resultado é liberada automaticamente ao final da execução do script.

10. Utilizando headers

O comando header permite enviar cabeçalhos html para o cliente. Deve ser utilizado por usuários que conheçam a função de cada header que pode ser enviado.

Sintaxe:

Header (string header);

Exemplo:

```
<?
    header (“location: login.php”);
?>
```

Não pode ser enviado depois de algum texto. veja o seguinte exemplo:

```
<html>
<? header("Location: http://www.php.net"); ?>
<body>
...
```

O código acima causará um erro, já que tentou-se enviar um header depois de ter sido enviado um texto(“<html>\n”).

Algumas utilizações do header são:

- redirecionar para outra página:

  Exemplo:

  header("Location: http://www.php.net");

- Definir o script como uma mensagem de erro:

  Exemplo:

  header("http/1.0 404 Not Found");

- Definir a expiração da página:

  Exemplos:

  header("Cache-Control: no-cache, must-revalidate"); // HTTP/1.1
  header("Pragma: no-cache"); // HTTP/1.0

11. Utilizando cookies

11.1. O que são Cookies

Cookies são variáveis gravadas no cliente(browser) por um determinado site. Somente o site que gravou o cookie pode ler a informação contida nele. Este recurso é muito útil para que determinadas informações sejam fornecidas pelo usuário apenas uma vez. Exemplos de utilização de cookies são sites que informam a quantidade de vezes que você já visitou, ou alguma informação fornecida numa visita anterior.

Existem cookies persistentes e cookies de sessão. Os persistentes são aqueles gravados em arquivo, e que permanecem após o browser ser fechado, e possuem data e hora de expiração. Os cookies de sessão não são armazenados em disco e permanecem ativos apenas enquanto a sessão do browser não for encerrada.

Por definição, existem algumas limitações para o uso de cookies, listadas a seguir:

- 300 cookies no total
- 4 kilobytes por cookie.
- 20 cookies por servidor ou domínio.

11.2. Gravando cookies

Para gravar cookies no cliente, deve ser utilizada a função setcookie.

Sintaxe:

setcookie(string nome, string valor, int exp, string path,string dominio, int secure);

onde:

- **nome** - nome do cookie;
- **valor** - valor armazenado no cookie;
- **exp** - data de expiração do cookie (opcional), no formato Unix. Se não for definida, o cookie será de sessão;
- **path** - path do script que gravou o cookie;
- **domínio** - domínio responsável pelo cookie;
- **secure** - se tiver valor 1, indica que o cookie só pode ser transmitido por uma conexão segura (https).

Observações:

- Um cookie não pode ser recuperado na mesma página que o gravou, a menos que esta seja recarregada pelo browser;
- Múltiplas chamadas à função setcookie serão executadas em ordem inversa;
- Cookies só podem ser gravados antes do envio de qualquer informação para o cliente. Portanto todas as chamadas à função setcookie devem ser feitas antes do envio de qualquer header ou texto.

11.3. Lendo cookies gravados

Os cookies lidos por um script PHP ficam armazenados em duas variáveis:

- No array $HTTP_COOKIE_VARS[], tendo como índice a string do nome do cookie;
- E numa variável cujo nome é o mesmo do cookie, precedido pelo símbolo $.

Exemplo:

Um cookie que foi gravado numa página anterior pelo seguinte comando:

setcookie("teste", "meu cookie");

Pode ser lida pela variável **$HTTP_COOKIE_VARS["teste"]** ou pela variável **$teste**.

O script a seguir é um exemplo do uso de cookies. O script exibe quantas vezes o usuário visitou aquela página.

Arquivo cookie.php

```
<?

    error_reporting(134);
    /*
     * Erro no script
     * Nome do seu Script
     * Por Seu nome ou nick
     * e-mail = seumail@qualquercoisa.com.br
     */

    ?>

    <?

    if ($Test != '') {
       $Test++;
       setcookie("Test",$Test, time()+3600000); // selecionar o nome do cokie
    } else {
       setcookie("Test",1, time()+3600000);  //Criar o valor 1
       $Test = 1;
    }
    $numvisits = $Test;

?>
```

Esse é o código completo do arquivo **cookie.php**. Agora, digite o arquivo **visitas.php** que irá informar o número de vezes que o site foi visitado por um mesmo usuário.

Arquivo visitas.php

```
<?
    include("cookie.php");

    echo "Você já esteve aqui $numvisits vezes";
?>
```

Agora, vamos estudar o código do arquivo **cookie.php** separadamente, para você entendê-lo melhor.

Logo após as informações pessoais e o reporting, podemos ver que temos o código:

```
<?
    if ($Test != ‘ ‘) {
        $Test++;
        setcookie("Test",$Test, time()+3600000);
    }
?>
```

Essa parte será a responsável pela criação do cookie no computador cliente (do usuário). O comando else que vem logo após o o código acima, é usado para acionar o outro código caso o cookie já exista no cliente.

Seguindo abaixo temos a fase terminal do script que respondera caso o cookie já exista, retornando o número de vezes que o usuário já esteve no site.

```
<?
    // ...
    } else {
       setcookie("Test",1, time()+3600000);
       $Test = 1;
    }
    $numvisits = $Test;
?>
```

Esse código após verificar o cookie irá responder quantas vezes o cliente já esteve no site e ambas as partes completarão a função de gravar novamente os novos dados no cookie.

Vá para o seu browser, carregue a página **visitas.php** e observe que o browser lhe informará o número de visitas que você fez no site.

Outro exemplo de uso de cookies:

```
<?
    setcookie("cookie", "teste", time() + 3600*24*365);
?>
```

Esse cookie tem validade de um ano:

3600 segundos = 1 hora
3600 segundos vezes 24 horas
24 horas vezes 365 dias

Para excluir um cookie basta fazer a mesmo sistema que criar ele, mas usando um prazo de expiração negativa. Veja o exemplo a seguir:

```
<?
  setcookie("cookie", "teste", time() – 3600*24*365);
?>
```

Veja acima o cookie que tinha validade de um ano será excluído pq foi passado que o tempo dele agora é de um ano atrás, ou seja, já acabou.

13. Trabalhando com Sessões (Sessions)

Uma sessão (**session**) é um período de tempo durante o qual uma pessoa navega num site. Uma sessão é gravada no servidor, e o tempo de duração de uma sessão pode ser ate o usuário fechar a página. Quando o usuário entra no site criamos uma sessão e ela recebe um número identificador único, chamado de session_id, para uma pagina ter o acesso aos dados da sessão ela tem que ter esse número identificador.

As sessões geralmente são usadas pra criar lojas virtuais e sistemas de login, onde o usuário entrará com usuário e senha em um formulário. Buscará no banco de dados e se achar algum usuário gravará na sessão uma identificação que dirá que ele já foi logado. Isso somente durará até o fechamento do browser.

Session_start() – é o comando para se utilizar uma sessão. Este comando deve ser **utilizado no início do código e antes de qualquer saída html.**

Sintaxe:

```
Session_start();
```

Exemplo:

```
<?
    session_start();
    $_SESSION["teste"] = 1;
?>
```

No exemplo acima, foi criado uma sessão com o nome se teste e valor 1. E nas paginas seguintes você pode restaurar esse valor assim:

```
<?
    session_start();
    echo $_SESSION["teste"];
?>
```

13.1. Função Session_register()

Você pode usar **session_register()** para registrar suas variáveis, no caso, o nome da variável na sessão será a mesma do nome da variável gravado, exemplo:

Exemplo:

```
<?
    $linguagem = "PHP";
    session_register("linguagem");
?>
```

Nesse caso foi gravada a variável $linguagem na variável de sessão “linguagem”, eis a vantagem de se usar $_SESSION que é recomendado a partir da versão 4.1, já que você poderá definir o nome da variável na sessão.

Obs.: session_register trabalha com register_globals on, e como está em extinção o uso em <u>on</u>, é sempre recomendado então usar $_SESSION, aliás, as facilidades são muito maiores, já que você trabalhará com elas como uma variável normal.

14. Funções do PHP

Neste capítulo iremos listar algumas das funções utilizadas pelo PHP. Procuramos citar as funções que verificamos serem comumente usadas por desenvolvedores de PHP. Para maiores informações sobre as funções aqui citadas e sobre as demais funções, consulte o manual do PHP no site: **www.php.net**.

14.1. Funções diversas

Função chr( )

Esta função retorna o caracter correspondente ao código ASCII fornecido.

Sintaxe:

```
string chr(int ascii);
```

Exemplo:

```
<?
    $texto = 65;
    $ascii = chr($texto);
    echo "O caracter correspondente ao código ".$texto." da Tabela ASCII é: "."<b>$ascii</b>";
?>
```

Função ord( ) - Esta função retorna o código ASCII correspondente ao caracter fornecido.

Sintaxe:

```
int ord(string string);
```

Exemplo:

```
<?
    $texto = 65;
    $ord = ord($texto);
    echo "O código da tabela ASCII correspondente ao caracter ".$texto." é: "."<b>$ord</b>";
?>
```

Função strlen( ) - Esta função retorna o tamanho da string fornecida.

Sintaxe:

```
int strlen(string str);
```

Exemplo:

```
<?
    $texto = “Curso de PHP";
    $retorno = strlen($texto);
    echo "Conteúdo de $texto: “.$texto<p>;
    echo “Quantidade de caracteres: "."<b>$retorno</b>";
?>
```

Função strtolower( ) - Esta função retorna a string fornecida com todas as letras minúsculas.

Sintaxe:

```
strtolower(string str);
```

Exemplo:

```
<?
    $texto = “CURSO DE  PHP"; // Digite o texto em letras maiúsculas.
    $retorno = strtolower($texto);
    echo "Conteúdo de $texto: “.$texto<p>;
    echo “Retorno da função: "."<b>$retorno</b>";
?>
```

Função strtoupper( ) - Esta função retorna a string fornecida com todas as letras maiúsculas.

Sintaxe:

```
string strtoupper(string str);
```

Exemplo:

```
<?
    $texto = “curso de  php"; // Digite o texto em letras minúsculas.
    $retorno = strtoupper($texto);
    echo "Conteúdo de $texto: “.$texto<p>;
    echo “Retorno da função: "."<b>$retorno</b>";
?>
```

Função ucfirst( ) - Esta função retorna a string fornecida com o primeiro caracter em letra maiúscula.

Sintaxe:

```
string ucfirst(string str);
```

Exemplo:

```
<?
    $texto = “curso de  php"; // Digite o texto em letras minúsculas.
    $retorno = ucfirst($texto);
    echo "Conteúdo de $texto: “.$texto<p>;
    echo “Retorno da função: "."<b>$retorno</b>";
?>
```

Função ucwords( ) - Esta função retorna a string fornecida com todas as palavras iniciadas por letras maiúsculas.

Sintaxe:

```
string ucwords(string str);
```

Exemplo:

```
<?
    $texto = “curso de  php"; // Digite o texto em letras minúsculas.
    $retorno = ucwords($texto);
    echo "Conteúdo de $texto: “.$texto<p>;
    echo “Retorno da função: "."<b>$retorno</b>";
?>
```

Função trim( ) - Esta função retira espaços e linhas em branco do início e do final da string fornecida.

Sintaxe:

```
string trim(string str);
```

Exemplo:

```
<?
    $texto = “   curso 01 ”;
    $retorno = trim($texto);
    echo “Retorno da função: "."<b>$retorno</b>";
?>
```

Função strstr( ) - Esta função procura a primeira ocorrência de **str2** em **str1**. Se não encontrar, retorna uma string vazia, e se encontrar retorna todos os caracteres de **str1** a partir desse ponto.

Sintaxe:

```
string strstr(string str1, string str2);
```

Exemplo:

```
<?
    texto = “Primeiro Curso de PHP;
    $retorno = strstr($texto, “Curso”);
    echo "Conteúdo de $texto: “.$texto<p>;
    echo “Retorno da função: "."<b>$retorno</b>";
?>
```

Função isset( ) - Esta função é utilizada para verificar se uma variável possui algum valor setado. Retorna **true** se a variável estiver setada (ainda que com uma string vazia ou o valor zero), e **false** em caso contrário.

Sintaxe:

```
int isset(mixed var);
```

Exemplo:

```
<?
    $texto = “curso”;
    $isset = isset($texto);
    echo "valor do isset: ".$isset;
?>
```

Função empty( ) - Esta função verifica se uma variável está vazia ou se contem o valor 0 (zero). Caso esteja vazia, retorna o valor true(1), caso contrário, retorna o valor false(0).

Sintaxe:

```
int empty(mixed var);
```

Exemplo:

```
<?
    $texto = “0”;
    $empty = empty($texto);
    echo "valor do isset: ".$isset;
?>
```

Função date( ) - Esta função retorna a data ou hora atual do sistema.

Sintaxe:

```
date("formato da data");
```

Exemplo:

```
<?
    $data = date("d/m/y");
    echo a data é: $data";
?>
```

A função date possui vários argumentos que serão utilizados de acordo com a forma como a data ou hora aparecerá para o usuário. Veja os exemplos no script a seguir.

Exemplo 2:

```
<?
    $dia = date('d');
    $mes = date('m');
    $ano2 = date('y');
    $ano4 = date('Y');
    $sem = date('w');
    $data = date('d/m/Y');
    $data1 = date('d/m/y');
    $hora = date('h:m:s');
    $h = date('h');
    $min = date("m");

    echo "
        Data atual exibindo o ano com 4 dígitos: <b>$data</b> e com 2 dígitos: <b>$data1.</b><p>
        \$dia = date('d')- exibe somente o dia da data atual do sistema. O dia é: <b>$dia</b><p>
        \$mes = date('m')- exibe somente o mês da data atual do sistema. O mês é:
        <b>$mes</b>$m<p>
        \$ano2 = date('y')- exibe somente o ano com 2 dígitos da data atual do sistema. O ano é:
        <b>$ano2</b><p>
        \$ano4 = date('Y')- exibe somente o ano com 4 dígitos da data atual do sistema. O ano é:
        <b>$ano4</b><p>
        \$sem = date('s')- exibe o número correspondente ao dia da semana, de 0(domingo) a
        7(sábado). O número é: <b>$sem</b><p>
        \$hora = date('h:m:s') - exibe a hora atual do sistema. A hora é: <b>$hora</b><p>
        \$h = date('h') - exibe somente o valor correspondente a hora do sistema. O valor da hora é:
        <b>$h</b><p>
        \$min = date('m') - exibe somente o valor correspondente aos minuntos da hora do sistema.
        O valor dos minutos é: <b>$min</b><p>
    ";
?>
```

Função explode( ) - Esta função retorna um array contendo partes da string fornecida separadas pelo padrão fornecido, sem limitar o número de elementos do array.

Sintaxe:

array explode(string padrao, string str);

Exemplo:

```
<?
    $data = “11/12/1975”;
    $data_array = explode(“/”,$data);
    $novadata = $xdata_array[2]."-".$xdata[1]."-".$xdata[0];
    echo $novadata;
?>
```

O código acima faz com que a variável **$data_array** receba o valor: array(11,12,1975);

O **PHP** irá exibir a data no seguinte formato:

Ano-mes-dia, ficando:

1975-12-11, pis o MySQL armazena as datas nesse formato.

Função nl2br( ) - Esta função retorna a string fornecida substituindo todas as quebras de linha (“\n”) por quebras de linhas em html (“<br>”).

A função acima pode ser utilizada, por exemplo, em um campo de comentário em um formulário.

Sintaxe:

string nl2br(string str);

Exemplo:

```
<?
    $data = date("d/m/y");
    echo nl2br("Curso de PHP\n Data de início: $data\n Valor: R$ 700,00\n");
?>
```

No exemplo acima, a função **nl2br** irá retornar:

Curso de PHP
Data de início: a data atual do seu sistema
Valor: R$ 700,00

Função substr( ) – Esta função retorna uma parte de uma string.

Sintaxe:

string substr (string string, int start [, int length])

Se start não for negativo, a string retornada iniciará na posição **start** em string, começando em zero. Por exemplo, na string 'abcdef', o caractere na posição 0 é 'a', o caractere na posição 2 é 'c', e assim em diante.

Exemplos:

Exemplo 1) Uso básico de substr()

```
<?
```

```
    $rest = substr("abcdef", 1); // retorna "bcdef"
    $rest = substr("abcdef", 1, 3); // retorna "bcd"
    $rest = substr("abcdef", 0, 4); // retorna "abcd"
    $rest = substr("abcdef", 0, 8); // retorna "abcdef"

    // Outra opção é acessar através de chaves

    $string = 'abcdef';
    echo $string{0}; // retorna a
    echo $string{3}; // retorna d
?>
```

Se **start** for negativo, a string retornada irá começar no caractere start a partir do fim de string.

Exemplo 2) Usando um inicio negativo

```
<?
    $rest = substr("abcdef", -1); // retorna "f"
    $rest = substr("abcdef", -2); // retorna "ef"
    $rest = substr("abcdef", -3, 1); // retorna "d"
?>
```

Se **length** for dado e for positivo, a string retornada irá conter **length** caracteres começando em **start** (dependendo do tamanho de string). Se a string é menor do que **start**, será retornado **FALSE**.

Se **length** for dado e for negativo, então esta quantidade de caracteres será omitida do final de string (após a posição de inicio ter sido calculada quando **start** for negativo).

Se **start** denota uma posição além da truncagem, uma string vazia será retornada.

Exemplo 3) Usando um length negativo

```
<?
    $rest = substr("abcdef", 0, -1); // retorna "abcde"
    $rest = substr("abcdef", 2, -1); // retorna "cde"
    $rest = substr("abcdef", 4, -4); // retorna ""
    $rest = substr("abcdef", -3, -1); // retorna "de"
?>
```

14.2. Funções de manipulação de arquivos

Função File_exists( ) - Esta função checa se um arquivo ou diretório existe no servidor.

Sintaxe:

bool file_exists (string nomedoarquivo)

Esta função retorna **TRUE** se o arquivo ou diretório especificado por nomedoarquivo existir; **FALSE** caso contrário.

Obs.: No Windows, para checar arquivos em compartilhamentos de rede, use:

- //computername/share/filename; ou
- \\\\computername\share\filename

Exemplo:

```
<?
    $filename = '/caminho/para/qualquer.txt';

    if (file_exists($filename)) {
        print "O arquivo $filename existe";
    } else {
        print "O arquivo $filename não existe";
    }
?>
```

Obs.: O resultado desta função é cacheada. Veja clearstatcache() para mais detalhes.

Esta função não trabalha com arquivos remotos, de forma que o arquivo a ser examinado precisa ser acessível pelo sistema de arquivos do servidor.

Função filesize( ) - Esta função é utilizada para ler o tamanho do arquivo. Retorna o tamanho do arquivo, ou FALSE em caso de erro.

Sintaxe:

int filesize (string nomedoarquivo)

Obs1.: O resultado desta função é cacheado. Veja clearstatcache() para mais detalhes.

Esta função não trabalha com arquivos remotos, de forma que o arquivo a ser examinado precisa ser acessível pelo sistema de arquivos do servidor.

Exemplo:

```
<?
    // Exibe algo como: arquivo.txt: 1024 bytes
    $filename = 'arquivo.txt';
    echo $filename . ': ' . filesize($filename) . ' bytes';
?>
```

Função filetype( ) - Esta função é utilizada para ler o tipo de arquivo.

É muito utilizada quando se deseja fazer uploads de arquivos e você precisa verificar se o tipo de arquivo que está sendo enviado está de acordo com os tipos permitidos por você. Arquivos do tipo **.exe**, **.ini**, **.bat**, normalmente são arquivos executáveis e que podem conter vírus. Por isso, na maioria dos casos, não são permitidos que sejam enviados para o servidor.

Esta função retorna o tipo do arquivo (file type). Os valores possíveis são fifo, char, dir, block, link, file e unknown (desconhecido). Retorna **FALSE** se um erro ocorrer. Esta função filetype() também produzirá um erro E_NOTICE se a chamada a stat falhar ou se o tipo de arquivo for desconhecido.

Sintaxe:

```
string filetype ( string nomedoarquivo)
```

Obs.: O resultado desta função é cacheado. Veja clearstatcache() para mais detalhes.

Esta função não trabalha com arquivos remotos, de forma que o arquivo a ser examinado precisa ser acessível pelo sistema de arquivos do servidor.

Exemplo:

```
<?
    echo filetype('/etc/passwd'); // file
    echo filetype('/etc/'); // dir
?>
```

Função: clearstatcache( ) - Esta função é utilizada para limpar as informações em cache de arquivos que foram verificados, por exemplo, pela função **filesize()**.

Sintaxe:

```
void clearstatcache ( void )
```

Quando você chama stat(), lstat() ou qualquer uma das funções afetadas (listadas abaixo), o PHP mantém em cache as informações que essas funções retornam para melhoria de performance. Entretanto, em certos casos você pode precisar limpar as informações cacheadas. Por exemplo, se um mesmo arquivo é verificado várias vezes em um único script, e esse arquivo corre o risco de ser apagado ou modificado durante a operação do script, você precisa limpar os dados do cache. Nesses casos, você pode utilizar a função clearstatcache() para limpar todas as informações que o PHP mantém sobre um arquivo.

As funções afetadas são stat(), lstat(), file_exists(), is_writable(), is_readable(), is_executable(), is_file(), is_dir(), is_link(), filectime(), fileatime(), filemtime(), fileinode(), filegroup(), fileowner(), filesize(), filetype(), e fileperms().

Obs.: Esta função guarda informações sobre arquivos específicos, de forma que você somente precisa chamar clearstatcache() se você estiver realizando várias operações sobre o mesmo arquivo e necessita que a informação sobre esse arquivo em particular não seja cacheada.

Exemplo:

```
<?
    echo filetype('/etc/passwd'); // file
    echo filetype('/etc/'); // dir
    clearstatcache(); // neste momento esta função limpou as informações do cachê do arquivo.
?>
```

PHP Avançado

Índice

9.1 - Arrays. 2
9.2 - Formulários. 21
9.3 - Arquivos. 32
9.4 - Diretórios. 56
9.5 - Path. 61
9.6 - Includes. 66
9.7 - Funções. 67
9.8 - Session. 78
9.9 - Cookies. 83
9.10 - Tratamento de Erros. 84
9.11 - Validação de Dados.. 95
9.12 - XML. 102
9.13 - Constantes Mágicas. 107
9.14 - Formatação. 109
9.15 - Imagens e Gráficos. 112
9.16 - Números. 126
9.17 - Permissões de Arquivos e Diretórios. 130
9.18 - Strings. 134
9.19 - URLs. 142
9.21 - Criptografia. 144
9.22 – E-mails.. 153
9.23 – Data e Hora. 157
Referências 178

9.1 - Trabalhando com Arrays em PHP

- 1 Trabalhando com Arrays
- 2 Algumas das funções
- 3 array_fill -- Preenche um array com valores
- 4 array_merge -- Funde dois ou mais arrays
- 5 array_pad -- Expande um array para um certo comprimento utilizando um determinado valor
- 6 array_pop -- Retira um elemento do final do array
- 7 array_push -- Adiciona um ou mais elementos no final de um array
- 8 array_reverse -- Retorna um array com os elementos na ordem inversa
- 9 array_search
- 10 array_shift -- Retira o primeiro elemento de um array
- 11 array_sum -- Calcula a soma dos elementos de um array
- 12 array -- Cria um array
- 13 arsort
- 14 asort
- 15 count -- Conta o número de elementos de uma variável
- 16 current -- Retorna o elemento corrente em um array
- 17 each -- Retorna o par chave/valor corrente de um array e avança o seu cursor
- 18 Percorrendo um array com each()
- 19 end -- Faz o ponteiro interno de um array apontar para o seu último elemento
- 20 key -- Retorna uma chave da posição atual de um array associativo
- 21 next -- Avança o ponteiro interno de um array
- 22 prev -- Retrocede o ponteiro interno de um array
- 23 sizeof -- Apelido de count()
- 24 sort -- Ordena um array pelo seu valor
- 25 Exemplo de array multidimensional
- 26 Exemplo de Array
- 27 Convertendo objetos para um array

Trabalhando com Arrays

Um array é uma variável, mas diferente das demais ele armazena uma coleção de valores e não somente um. E ainda por cima podem conter outras variáveis e de tipos diferentes.

Detalhe importante: Quando em uma função precisarmos rtornar mais de um valor array é a saída, basta retornar todos os valores em forma de array.

Além disso é semelhante ao que estudamos na matemática: linhas e colunas. Matriz 3x4 (3 linhas e 4 colunas).

Um array no PHP é um mapa ordenado, que relaciona valores com chaves (em linhas e colunas).

Especificando um array()

array([chave =>] valor, ...);

A chave pode ser uma string ou um inteiro. O valor pode ser qualquer coisa.

Algumas das funções

Essas funções abaixo permitem a interação e manipulação de arrays de várias formas. Arrays são essenciais para armazenar, gerenciar, operar sobre um conjunto de variáveis.

Arrays (matrizes) simples e multidimensionais são suportados e podem ser criados pelo usuário ou por outras funções. Existem diversas funções específicas para bancos de dados, que preenchem arrays com os dados retornados em consultas, e vários outros tipos de funções também retornam arrays.

array_fill -- Preenche um array com valores

array array_fill (int start_index, int num, mixed value)

```
<?php
$a = array_fill(5, 6, 'banana');
print_r($a);
?>
```

array_merge -- Funde dois ou mais arrays

array array_merge (array array1, array array2 [, array ...])

```
<?php
$array1 = array();
$array2 = array(1 => "data");
$result = array_merge($array1, $array2);
?>
```

Não esqueça que as chaves numéricas serão reordenadas!

```
Array
(
   [0] => data
)
```

Se você quer preservar os arrays e apenas concatená-los, o operador +:

```
<?php
$array1 = array();
$array2 = array(1 => "data");
$result = $array1 + $array2;
?>
```

As chaves numéricas serão preservadas e as associações originais permanecem.

array_pad -- Expande um array para um certo comprimento utilizando um determinado valor

array array_pad (array input, int pad_size, mixed pad_value)

Exemplo 1. Exemplo de array_pad()

```
<?php
$input = array(12, 10, 9);

$result = array_pad($input, 5, 0);
// $result é array(12, 10, 9, 0, 0)

$result = array_pad($input, -7, -1);
// $result é array(-1, -1, -1, -1, 12, 10, 9)

$result = array_pad($input, 2, "noop");
// Não será expandido.
?>
```

array_pop -- Retira um elemento do final do array

mixed array_pop (array array)

```
<?php
$cesta = array("laranja", "banana", "melancia", "morango");
$fruta = array_pop($cesta);
print_r($cesta);
?>
```

array_push -- Adiciona um ou mais elementos no final de um array

int array_push (array array, mixed var [, mixed ...])

```
<?php
$cesta = array("laranja", "morango");
array_push($cesta, "melancia", "batata");
print_r($cesta);
?>
```

array_reverse -- Retorna um array com os elementos na ordem inversa

array array_reverse (array array [, bool preserve_keys])

```
<?php
$input = array("php", 4.0, array ("verde", "vermelho"));
```

```
$result = array_reverse($input);
$result_keyed = array_reverse($input, TRUE);
print_r($result_keyed);
?>
```

array_search

-- Procura por um valor em um array e retorna sua chave correspondente caso seja encontrado. Caso contrário retorna FALSE.

mixed array_search (mixed procurar_este, array procurar_neste [, bool strict])

```
<?php
$a=array("a","b",0,"c","d");
echo "a: ".array_search("a",$a)."<br>";
echo "b: ".array_search("b",$a)."<br>";
echo "c: ".array_search("c",$a)."<br>";
echo "d: ".array_search("d",$a)."<br>";
echo "0: ".array_search("0",$a)."<br>";
echo "x: ".array_search("x",$a)."<br>";
echo "1: ".array_search("1",$a);
?>

<?php
if (array_search($needle, $array)!== FALSE) {
   //code goes here (
}
?>

<?php

function array_replace($search, $replace, &$array) {
  foreach($array as $key => $value) {
   if($value == $search) {
     $array[$key] = $replace;
   }
  }
}
?>

<?
$Projects[0] = array(123, "Text 1");
$Projects[1] = array(456, "Text 2");
$Projects[2] = array(789, "Text 3");

$search_value = "ext 3";

foreach ($Projects as $key => $row){
     foreach($row as $cell){
       if (strpos($cell, $search_value) !== FALSE){
                echo  "<p>Project ".$key;
       }
     }
 }
?>
```

array_shift -- Retira o primeiro elemento de um array

mixed array_shift (array array)

```
<?php
$cesta = array("laranja", "banana", "melancia", "morango");
$fruta = array_shift($cesta);
print_r($cesta);
?>
```

array_sum -- Calcula a soma dos elementos de um array

mixed array_sum (array arr)

```
<?php
$a = array(2, 4, 6, 8);
echo "soma(a) = ".array_sum($a)."<br>";

$b = array("a" => 1.2, "b" => 2.3, "c" => 3.4);
echo "soma(b) = ".array_sum($b)."<br>";
?>
```

array_unique -- Remove os valores duplicados de um array

array array_unique (array array)

```
<?php
$input = array("a" => "verde", "vermelho", "b" => "verde", "azul", "vermelho");
$result = array_unique($input);
print_r($result);
?>
```

Exemplo 2. array_unique() e tipos

```
<?php
$input = array(4, "4", "3", 4, 3, "3");
$result = array_unique($input);
var_dump($result);
?>
<pre>


<h2>array_values -- Retorna todos os valores de um array</h2>

array array_values ( array input )

//Retorna os valores, as chaves não
<pre>
<?php
$array = array("tamanho" => "G", "cor" => "dourado");
print_r(array_values ($array));
?>
```

array -- Cria um array

array array ([mixed ...])

Exemplo 1. Exemplo de array()

```
<?php
$frutas = array (
   "frutas" => array("a"=>"laranja", "b"=>"banana", "c"=>"maçã"),
   "numeros" => array(1, 2, 3, 4, 5, 6),
   "buracos" => array("primeiro", 5 => "segundo", "terceiro")
)
?>
```

Exemplo 2. Indexação automática com array()

```
<?php
$array = array(1, 1, 1, 1,  1, 8 => 1,  4 => 1, 19, 3 => 13);
print_r($array);
?>
```

arsort

-- Ordena um array em ordem descrescente dos valores mantendo a associação entre índices e valores

void arsort (array array [, int sort_flags])

```
<?php
$frutas = array("d" => "limao", "a" => "laranja", "b" => "banana", "c" =>
"melancia");
arsort($frutas);
reset($frutas);
while (list($chave, $valor) = each($frutas)) {
   echo "$chave = $valor\n";
}
?>
```

asort

-- Ordena um array em ordem crescente dos valores mantendo a associação entre índices e valores

void asort (array array [, int sort_flags])

```
<?php
$frutas = array("d" => "limao", "a" => "laranja", "b" => "banana", "c" =>
"melancia");
asort($frutas);
reset($frutas);
```

```
while (list($chave, $valor) = each($frutas)) {
   echo "$chave = $valor\n";
}
?>
```

count -- Conta o número de elementos de uma variável

int count (mixed var [, int mode])

```
<?php
$a[0] = 1;
$a[1] = 3;
$a[2] = 5;
$a[3] = 6;
$result = count($a);
// $result == 4
print $result."<br>";

$b[0] = 7;
$b[5] = 9;
$b[10] = 11;
$result = count($b);
// $result == 3;
print $result;
?>
```

Exemplo 2. Uso recursivo da função count() (PHP >= 4.2.0)

```
<?php
$food = array( 'fruits'  => array('orange', 'banana', 'apple'),
'veggie'  => array('carrot', 'collard','pea'));
// recursive count
echo count($food,COUNT_RECURSIVE);  // mostra 8
// normal count
echo count($food);                  // mostra2 2
?>

<?php
$food = array( 'fruits'  => array('orange', 'banana', 'apple'),
'veggie'  => array('carrot', 'collard','pea'));
// recursive count
echo count($food,COUNT_RECURSIVE)."<br>";  // mostra 8
// normal count
echo count($food);                  // mostra2 2
?>
```

current -- Retorna o elemento corrente em um array

mixed current (array array)

```
<?php
```

```
$transport = array('foot', 'bike', 'car', 'plane');
$mode = current($transport); // $mode = 'foot';
echo "Atual $mode<br>";
$mode = next($transport);    // $mode = 'bike';
echo "Atual $mode<br>";
$mode = current($transport); // $mode = 'bike';
echo "Atual $mode<br>";
$mode = prev($transport);    // $mode = 'foot';
echo "Atual $mode<br>";
$mode = end($transport);    // $mode = 'plane';
echo "Atual $mode<br>";
$mode = current($transport); // $mode = 'plane';
echo "Atual $mode<br>";
?>
```

each -- Retorna o par chave/valor corrente de um array e avança o seu cursor

array each (array array)

```
<?php
$foo = array("bob", "fred", "jussi", "jouni", "egon", "marliese");
$bar = each($foo);
print_r($bar);
?>

<?php
  $foo = array("Robert" => "Bob", "Seppo" => "Sepi");
  $bar = each($foo);
  print_r($bar);
?>
```

Percorrendo um array com each()

```
<?php
$fruit = array('a' => 'apple', 'b' => 'banana', 'c' => 'cranberry');
reset($fruit);
while (list($key, $val) = each($fruit)) {
     echo "$key => $val\n";
}
/* Saída:

  a => apple
  b => banana
  c => cranberry

  */
?>
```

end -- Faz o ponteiro interno de um array apontar para o seu último elemento

mixed end (array array)

```
<?php
  $frutas = array('melancia', 'banana', 'morango');
  print end($frutas); // morango
?>
```

key -- Retorna uma chave da posição atual de um array associativo

mixed key (array array)

```
<?php
  $array = array(
     'fruit1' => 'apple',
     'fruit2' => 'orange',
     'fruit3' => 'grape',
     'fruit4' => 'apple',
     'fruit5' => 'apple');

  // este ciclo  exibirá todas as chaves do array associativo
  // ausiliado pela função next()
  while ($fruit_name = current($array)) {
         echo key($array).'<br>';
             next($array);
  }
  ?>

<?php
  $array = array(
     'fruit1' => 'apple',
     'fruit2' => 'orange',
     'fruit3' => 'grape',
     'fruit4' => 'apple',
     'fruit5' => 'apple');

  // este ciclo  exibirá toda a chave do array associativo
  // onde o valor é igual a "apple"
  while ($fruit_name = current($array)) {
     if ($fruit_name == 'apple') {
         echo key($array).'<br>';
     }
     next($array);
  }
  ?>
```

next -- Avança o ponteiro interno de um array

mixed next (array array)

```
<?php
  $transport = array('foot', 'bike', 'car', 'plane');
  $mode = current($transport); // $mode = 'foot';
  print"$mode<br>";
  $mode = next($transport);    // $mode = 'bike';
  print"$mode<br>";
  $mode = next($transport);    // $mode = 'car';
  print"$mode<br>";
  $mode = prev($transport);    // $mode = 'bike';
  print"$mode<br>";
  $mode = end($transport);    // $mode = 'plane';
  print"$mode<br>";
?>
```

prev -- Retrocede o ponteiro interno de um array

mixed prev (array array)

```
<?php
  $transport = array('foot', 'bike', 'car', 'plane');
  $mode = current($transport); // $mode = 'foot';
print"$mode<br>";
  $mode = next($transport);    // $mode = 'bike';
print"$mode<br>";
  $mode = next($transport);    // $mode = 'car';
print"$mode<br>";
  $mode = prev($transport);    // $mode = 'bike';
print"$mode<br>";
  $mode = end($transport);    // $mode = 'plane';
print"$mode<br>";
?>
<pre>


<h2>reset --  Faz o ponteiro interno de um array apontar para o seu primeiro
elemento</h2>

mixed reset ( array array )
<pre>
<?php
  $array = array('primero passo', 'segundo passo', 'terceiro passo', 'quarto
passo');

  // por definição, o ponteiro está sobre o primeiro elemento
  echo current($array)."<br>\n"; // "Primeiro passo"

  // pula dois passos
  next($array);
  next($array);
  echo current($array)."<br>\n"; // "passo três"
```

```
  // reinicia o ponteiro, começa novamente o primeiro passo
  reset($array);
  echo "Depois de resetado...: " . current($array)."<br>\n"; // "primeiro passo"
?>
```

sizeof -- Apelido de count()

sort -- Ordena um array pelo seu valor

void sort (array array [, int sort_flags])

```
<?php

$frutas = array("limao", "laranja", "banana", "melancia");
sort($frutas);
reset($frutas);
while (list($chave, $valor) = each($frutas)) {
   echo "frutas[".$chave."] = ".$valor."<br>";
}
?>
```

Os seguintes também são funcionalmente idênticos:

```
<?php
$arr = array("one", "two", "three");
reset($arr);
while (list($key, $value) = each ($arr)) {
   echo "Chave: $key; Valor: $value<br />\n";
}

foreach ($arr as $key => $value) {
   echo "Chave: $key; Valor: $value<br />\n";
}
?>
```

Mais alguns exemplos para demonstrar os usos:

```
<?php
/* exemplo foreach 1: somente valores */

$a = array(1, 2, 3, 17);

foreach ($a as $v) {
   echo "Valor atual de \$a: $v.\n";
}

/* exemplo foreach 2: valores (com as chaves impressas para ilustração) */

$a = array(1, 2, 3, 17);
```

```
$i = 0; /* para exemplo somente */

foreach ($a as $v) {
   echo "\$a[$i] => $v.\n";
   $i++;
}

/* exemplo foreach 3: chaves e valores */

$a = array (
   "um" => 1,
   "dois" => 2,
   "três" => 3,
   "dezessete" => 17
);

foreach ($a as $k => $v) {
   echo "\$a[$k] => $v.\n";
}

/* exemplo foreach 4: arrays multidimensionais */

$a[0][0] = "a";
$a[0][1] = "b";
$a[1][0] = "y";
$a[1][1] = "z";

foreach ($a as $v1) {
   foreach ($v1 as $v2) {
       echo "$v2\n";
   }
}

/* exemplo foreach 5: arrays dinâmicos */

foreach (array(1, 2, 3, 4, 5) as $v) {
   echo "$v\n";
}
?>
```

Exemplo de array multidimensional

```
$produto[1][codigo] = "1";
$produto[1][nome] = "João Pereira Brito";
$produto[1][email] = "joao@joao.org";
$produto[1][rua] = "Vasco da Gama";
$produto[1][numero] = "1345";

$produto[2][codigo] = "2";
$produto[2][nome] = "Antônio queiroz";
```

Exemplo de Array

```
$i=0;
while($i < $numregs){
        $codigo=pg_result($consulta,$i,codigo);
        $nome=pg_result($consulta,$i,nome);
        $venc=pg_result($consulta,$i,vencimento);
        $apartamento=pg_result($consulta,$i,apartamento);
        $pessoas=pg_result($consulta,$i,pessoas);
        $cota_agua=pg_result($consulta,$i,cota_agua);
        $cota_condominio=pg_result($consulta,$i,cota_condominio);
        $cota_reserva=pg_result($consulta,$i,cota_reserva);

        $total = $cota_agua + $cota_condominio + $cota_reserva;
        $total = number_format($total,2, ',','.');
...
$i++;
}
```

Também podemos ter um array formado por outros arrays (neste caso, cada sub array é uma linha do principal)

```
$arrayvarios = array(
        array(1, 3, 5, 7),
        array(2, 4, 6, 8),
        array(1, 1, 1, 1)
);
```

Neste caso temos um array 2x4 (2 linhas por 4 colunas, que iniciam sempre com índice zero).

Então se queremos retornar o valor 8, que está na linha 2 e coluna 4, devemos retornar o índice 1,3 (linha2=índice 1, coluna4=índice3).

print $arrayvarios[1][3];

Agora veremos com detalhes os pares: chave => valor:

```
$alunos = array(
        "0732355" => "Ribamar FS",
        "0823456" => "Antônio Brito",
        "0654345" => "Roberto Queiroz"
);
```

O que isto retornaria?

```
print $alunos["0732355"];
print $alunos[0];
```

Experimente!!

Atribuindo valores às chaves de arrays

Também podemos fazer diretamente assim:

print $alunos["0732355"] = "João Brito";

Lembrando que, a chave, é exclusiva. Podemos ter

```
$alunos["0732355"] = "João Brito";
```

```
$alunos["0932355"] = "João Brito";

Mas não podemos ter:
$alunos["0732355"] = "João Brito";
$alunos["0732355"] = "Ribamar FS";
```

Anexo agora um excelente tutorial sobre Arrays do Celso Goya publicado em:
http://www.xoopstotal.com.br/modules/wfsection/article.php?articleid=51

Trabalhando com arrays

Visão geral Para facilitar o entendimento, vamos definir array como um conjunto de valores, que podem ser identificados em grupo ou então separadamente. Estes conjuntos podem ser muito úteis enquanto programamos, pois em alguns casos podem substituir uma tabela em banco de dados ou então utilizando métodos mais avançados podemos carregá-los dinâmicamente e utilizar quase como um banco de dados em memória.

A linguagem PHP oferece uma incrível gama de recursos para se trabalhar com arrays. Com destaque para as funções auxiliares que permitem fazer desde uma simples contagem de elementos até a conversão automática de um array em string.

Neste artigo desenvolveremos como exemplo uma função para gerar combo boxes com os estados do Brasil. Muitas vezes criamos uma tabela no banco de dados para armazenar a lista de estados do Brasil sendo que neste caso existe um número finito de registros e menor que 100, então as operações de banco de dados não são tão ágeis quanto o uso de um array.

Criando o primeiro array

Para utilizar um array, antes de mais nada é preciso criar uma variável do tipo array.

```
<?php
$estados = array();
?>
```

O próximo passo é montar nossa lista de estados.

```
<?php
$estados = array();
$estados[0] = "Acre";
$estados[1] = "Alagoas";
$estados[2] = "Amapá";
$estados[3] = "Amazonas";
?>
```

Os colchetes servem para identificar qual elemento do nosso conjunto estamos nos referindo e o número entre colchetes é o código identificador do elemento.

Podemos fazer o seguinte teste:

```
<?php
$estados = array();
$estados[0] = "Acre";
$estados[1] = "Alagoas";
```

```
$estados[2] = "Amapá";
$estados[3] = "Amazonas";
echo($estados[0]);
?>
```

Neste caso será exibida a palavra Acre, pois indicamos o ítem [0] da variável $estados, que é um array.

Você deve estar se perguntando "O que há de tão fantástico em um array?". Agora vamos mostrar alguns recursos.

Criando o array de estados

Nosso array não será de grande valia se não permitir que as siglas dos estados sejam armazenadas também, pois desta forma podemos guardar no banco de dados apenas os dois caractéres correspondentes à sigla do estado, ou seja, utilizaremos apenas dois bytes no banco de dados.

Então vamos criar um array com duas colunas, sendo a primeira a sigla do estado e a segunda seu nome por extenso.

```
<?php
$estados = array();
$estado[0][0] = "AC";
$estado[0][1] = "Acre";
$estado[1][0] = "AL";
$estado[1][1] = "Alagoas";
$estado[2][0] = "AP";
$estado[2][1] = "Amapá";
$estado[3][0] = "AM";
$estado[3][1] = "Amazonas";
$estado[4][0] = "BA";
$estado[4][1] = "Bahia";
$estado[5][0] = "CE";
$estado[5][1] = "Ceará";
$estado[6][0] = "DF";
$estado[6][1] = "Distrito Federal";
$estado[7][0] = "ES";
$estado[7][1] = "Espírito Santo";
$estado[8][0] = "GO";
$estado[8][1] = "Goiás";
$estado[9][0] = "MA";
$estado[9][1] = "Maranhão";
$estado[10][0] = "MG";
$estado[10][1] = "Minas Gerais";
$estado[11][0] = "MT";
$estado[11][1] = "Mato Grosso";
$estado[12][0] = "MS";
$estado[12][1] = "Mato Grosso do Sul";
$estado[13][0] = "PA";
$estado[13][1] = "Pará";
$estado[14][0] = "PR";
$estado[14][1] = "Paraná";
$estado[15][0] = "PE";
$estado[15][1] = "Pernambuco";
```

```
$estado[16][0] = "PI";
$estado[16][1] = "Piauí";
$estado[17][0] = "RJ";
$estado[17][1] = "Rio de Janeiro";
$estado[18][0] = "RN";
$estado[18][1] = "Rio Grande do Norte";
$estado[19][0] = "RS";
$estado[19][1] = "Rio Grande do Sul";
$estado[20][0] = "RO";
$estado[20][1] = "Rondônia";
$estado[21][0] = "RR";
$estado[21][1] = "Roraima";
$estado[22][0] = "SC";
$estado[22][1] = "Santa Catarina";
$estado[23][0] = "SP";
$estado[23][1] = "São Paulo";
$estado[24][0] = "SE";
$estado[24][1] = "Sergipe";
$estado[25][0] = "TO";
$estado[25][1] = "Tocantins";
?>
```

A diferença neste exemplo é que utilizamos dois identificadores de elemento, ou seja, agora para cada elemento do array possuimos mais outros dois dependentes. Da mesma forma que criamos dois elementos o 0 e 1 para cada item de estado poderiamos criar n novos sub-elementos, por exemplo:

```
<?php
$estado = array();

$estado[0][0] = "SP";
$estado[0][1] = "São Paulo";
$estado[0][2] = "Sudeste";
?>
```

Vamos considerar à partir de agora que um array possui linhas e colunas, onde as linhas são equivalentes ao primeiro conjunto de colchetes e as colunas são equivalentes ao segundo conjunto de colchetes.

A função de exibição do combo box de estados

Agora vamos exibir todos os elementos de nosso array em uma função:

```
<?php
/*Nossa função reebe 3 parâmetros
$pNome :: Corresponde ao nome do SELECT
$pSelected :: Corresponde ao elemento que deverá possuir o status de selecionado
automaticamente
$extra :: Caso precise adicionar um style, ou então opção de multiple
*/
function renderCombo($pNome = "", $pSelected = "SP", $extra = ""){
        echo("<SELECT NAME='".$pNome."' ".$extra.">");

/*
Para exibir todos os itens do nosso combo utilizamos o comando for ,
lembre-se que como usamos números para identificar nosso array,
então podemos substituí-lo automaticamente com o for
```

```
*/
        //Realiza o loop em todos os elementos do array
        for( $i = 0; $i < 26;$i++ ){
                //Imprime a TAG OPTION usando a primeira coluna do array
                echo("<OPTION VALUE='".$estado[$i][0]."'");
                //Efetua a comparação para verificar se este é o ítem
selecionado
                if( $estado[$i][0] == $pSelected ){
                        //Caso a comparação seja verdadeira seleciona o ítem
                        echo(" SELECTED");
                }
                //Imprime o nome por extenso do estado, equilavente a segunda
coluna do array
                echo(">".$estado[$i][1]."</option>\n");
        }
        //Finaliza a tag SELECT
        echo("</SELECT>\n");
}
?>
```

Eureka! Esta feita uma função para exibir um combo de estados.

Identificadores alternativos

Na linguagem PHP podemos utilizar palavras para identificar um elemento de um array, este recurso é muito bom, pois facilita muito a depuração e o entendimento de programas que utilizam arrays.

Vamos utilizar nosso array de estados como exemplo:

```
<?php
$estado[0]["sigla"] = "SP";
$estado[0]["nome"] = "São Paulo";
$estado[0]["regiao"] = "Sudeste";

echo($estado[0]["sigla"]);
?>
```

Desta forma podemos deixar o código de nossos programas mais fáceis de se compreender. Repare que utilizamos uma string simples para identificar um elemento do array, sendo assim, podemos utilizar variáveis para identificar um item do array, por exemplo:

```
<?php
$estado[0]["sigla"] = "SP";
$estado[0]["nome"] = "São Paulo";
$estado[0]["regiao"] = "Sudeste";

$variavel = "sigla";

echo($estado[0][$variavel]);
?>
```

É importante lembrar que mesmo existindo uma string para identificar um elemento do array ainda podemos utilizar números se quisermos, por exemplo:

```
<?php
$estado[0]["sigla"] = "SP";
$estado[0]["nome"] = "São Paulo";
$estado[0]["regiao"] = "Sudeste";

echo($estado[0][0]);
?>
```

Nos três casos o resultado é o mesmo, diferindo apenas no método como chamamos o array.

O que você viu neste artigo é o básico sobre arrays, caso você se interessar pelo assunto e queira dar uma pesquisada rápida na web, vai encontrar outras formas de declarar arrays bem como usos diferenciados. O XOOPS utiliza muito este recurso. É só dar uma olhada em algum arquivo xoops_version.php, que você vai encontrar um exemplo prático do uso de arrays.

Final do tutorial do Celso Goya.

Convertendo objetos para um array

http://www.revistaphp.com.br/print.php?id=147

```
$array = array();

if(is_array($array)){
    echo 'Variável $array é um array';
} else {
    echo 'Variable is not an array';
}
```

Usando Array em Laços

For

```
<?php

 $animals = array( 'dingo', 'wombat', 'Steve Irwin', 'playpus', 'emu' );

 /*** Recebe o tamanho do array ***/
 $size = sizeof( $animals );

 for( $i=0; $i<$size; $i++ ){
      echo $animals[$i].'<br />';
 }

?>
```

While

```
<?php

 $animals = array( 'dingo', 'wombat', 'Steve Irwin', 'playpus', 'emu' );

 $size = sizeof( $animals );

 /*** set um contador ***/
 $i = 0;

 while( $i < $size )
 {
       echo $animals[$i].'<br />';
       $i++;
 }

?>
```

foreach
O foreach varre todos os elementos de um array

```
<?php

 $animals = array( 'dingo', 'wombat', 'Steve Irwin', 'playpus', 'emu' );

 foreach( $animals as $animal )
 {
       echo $animal.'<br />';
 }

?>
```

Adicionar um Elemento para um Array
Usar a função array_push().

```
<?php

$animals = array(
       'ernie'=>'dingo',
       'wally'=>'wombat',
       'rat bag'=>'Steve Irwin',
       'playto'=>'platypus',
       'marty'=>'emu'
       );

 /*** Adicionar um novo elemento para o array ***/
 array_push( $animals, 'wallaby', 'stingray' );
```

```
 foreach( $animals as $key=>$animal ){
      echo $key.' - '.$animal.'<br />';
 }

?>
```

Excluir elemento de array
Usar a função unset()

```
<?php

  /*** create an array ***/
  $animals = array(
      'ernie'=>'dingo',
      'wally'=>'wombat',
      'rat bag'=>'Steve Irwin',
      'playto'=>'platypus',
      'marty'=>'emu'
      );

 /*** delete an array element ***/
 unset( $animals['rat bag'] );

 /*** loop over the array ***/
 foreach( $animals as $key=>$animal )
 {
      echo $key.' - '.$animal.'<br />';
 }

?>
```

9.2 - Trabalhando com Formulários em PHP

Manipulando dados de formulários com PHP – Parte 1

Uma das dúvidas mais freqüentes entre programadores PHP iniciantes é como manipular os dados de formulário enviados para os scripts PHP, principalmente dados de "checkbox" e upload de arquivos. Nessa primeira parte desse artigo, estarei mostrando como receber e manipular dados de campos comuns de formulários.

Na próxima semana estarei mostrando como manipular o upload de arquivos através dos formulários.

Para facilitar, esta primeira parte está divida nos seguintes tópicos:

1 - Introdução
2 - Campos Hidden
3 - Campos Text e Textarea
4 - Campos Radio
5 - Campos Checkbox
6 - Campos Select

1 - Introdução

Um formulário HTML é apenas um "rosto bonito" para onde os usuários poderão inserir informações que serão interpretados de alguma maneira por algum script do lado do servidor. E no nosso caso, esse script é um script PHP.

Primeiro: antes para poder enviar as informações, seu formulário deve conter um botão "submit", isso se consegue através do comando:

```
<input type=submit value="Texto do Botão">
```

Segundo: todos os campos que serão tratados no script PHP devem conter o parâmetro "NAME", caso contrário, os dados não serão passados para o script PHP. Ex: <input type=text name=nome_do_campo>

Como as informações do formulário são passadas para esse script PHP e como as informações do formulário enviado são tratadas, dependem de você.

Existem 2 métodos como as informações podem ser passadas: GET e POST. O recomendável sempre, para todos os formulários é usar o método POST, onde os dados enviados não são visíveis nas URLs, ocultando possíveis importantes informações e permitindo o envio de longas informações. O GET é totalmente o contrário disso.

Como as informações chegam para o script PHP?

Assuma o seguinte formulário:

```
<form action="script.php" method="post">
Campo 1: <input type=text name=campo1><br>
Campo 2: <input type=text name=campo2><br>
<input type=submit value="OK">
</form>
```

Esse formulário usa o método POST para envio das informações, então em "script.php":

```
<?php
echo "O valor de CAMPO 1 é: " . $_POST["campo1"];
echo "<br>O valor de CAMPO 2 é: " . $_POST["campo2"];
?>
```

Se o formulário tivesse sido enviado usando o método GET, você simplesmente usaria $_GET no lugar de $_POST.

Observações:

Em vez de usar $_GET ou $_POST você pode escrever a variável com o mesmo nome do campo do formulário (no exemplo, $campo1 e $campo2). Mas, esse uso não é recomendado, pois se a diretiva "register_globals" na configuração do seu PHP estiver desativada, as variáveis com nome dos campos dos formulários, terão um valor vazio.

Uma solução para isso é usar a função **import_request_variables** no começo dos seus scripts que interpretam formulários. Essa função aceita 3 letras como argumentos: P, G e C, referentes a $_POST, $_GET e $_COOKIE respectivamente. Exemplo de uso:

```
<?php
import_request_variables("gP");
?>
```

O que acontece?

Exemplo: Você possui formulário com os campos "nome", "endereço" e "idade". Assuma que a diretiva "register_globals" do seu PHP esteja desligada, mas, você já havia programado o script usando as variáveis no escopo global, no lugar de $_POST.

Adicionando aquela função no começo do script, as variáveis do seu formulário postado: $_POST["nome"], $_POST["endereco"] e $_POST["idade"] serão extraídas cada para uma variável diferente: $nome, $endereco e $idade.

2 - Campos Hidden

Os campos hidden são usados para passar informações que não podem ser alteradas pelo usuário que estará inserindo informações no formulário. Por exemplo: você tem um site com sistema de login e o usuário quer alterar as informações de login dele. O script que irá manipular esse formulário, precisa saber o ID do usuário para poder alterar as informações no banco de dados,

então esse ID é um campo hidden.

Códigos Exemplos:

hidden.html

```
<form action="hidden.php" method="post">
<input type=hidden name=escondido value="valor do escondido">
<input type=hidden name=id value="111">
<input type=submit>
</form>
```

hidden.php

```
<?php
echo "Campo Hidden: " . $_POST["escondido"];
echo "<br>Oi, seu ID é: " . $_POST["id"];
?>
```

3 - Campos Text e Textarea

Os campos text e textarea são os tipos mais simples, onde há somente um possível valor por campo. Dispensam maiores explicações.

Códigos Exemplos:

texts.html

```
<form action="texts.php" method="post">
Nome: <input type=text name=nome><br>
Email: <input type=text name=email><br><br>
Mensagem: <textarea name=mensagem cols=8 rows=3></textarea><br>
<input type=submit>
</form>
```

texts.php

```
<?php
echo "Olá " . $_POST["nome"] . " (email: " . $_POST["email"] . ")<br><br>";
echo "Sua mensagem: " . $_POST["mensagem"];
?>
```

4 - Campos Radio

Campos Radio permitem um relacionamento de um para muitos entre identificador e valor, ou seja, eles têm múltiplos possíveis valores, mas somente um pode ser pré-exibido ou selecionado. Por exemplo: você tem um sistema de "quiz". Cada pergunta possui 5 possíveis respostas. Cada resposta é um radio, onde os 5 radios dessa pergunta possuem o mesmo identificador, mas cada com valores diferentes.

Códigos Exemplos:

radio.html

```
<form action="radio.php" method="post">
<B>Qual seu sistema operacional?</B><br>
<input type=radio name=sistema value="Windows 98"> Win 98
<input type=radio name=sistema value="Windows XP"> Win XP
<input type=radio name=sistema value="Linux"> Linux
<input type=radio name=sistema value="Mac"> Mac
<br><br>
<B>Qual a marca de seu monitor?</B><br>
<input type=radio name=monitor value="Samsung"> Samsung
<input type=radio name=monitor value="LG"> LG
<input type=radio name=monitor value="Desconhecido"> Desconhecido
<br><br>
<input type=submit>
</form>
```

radio.php

```
<?php
echo "Seu sistema operacional é: " . $_POST["sistema"];
echo "<br>Seu monitor é: " . $_POST["monitor"];
?>
```

5 - Campos Checkbox

O tipo Checkbox tem somente um possível valor por entrada: on value (marcado) ou no value (desmarcado). No script você deve fazer a verificação para saber se o campo foi marcado ou não.

Se é possível também utilizar grupos de checkbox com o mesmo nome. Para você deve adicionar "[]" no final do nome, para o PHP interpretar como array, veja o código exemplo.

Códigos Exemplos:

checkbox.html

```
<form action="checkbox.php" method="post">
<B>Escolha os numeros de sua preferência:</B><br>
<input type=checkbox name="numeros[]" value=10> 10<br>
<input type=checkbox name="numeros[]" value=100> 100<br>
<input type=checkbox name="numeros[]" value=1000> 1000<br>
<input type=checkbox name="numeros[]" value=10000> 10000<br>
<input type=checkbox name="numeros[]" value=90> 90<br>
<input type=checkbox name="numeros[]" value=50> 50<br>
<input type=checkbox name="numeros[]" value=30> 30<br>
<input type=checkbox name="numeros[]" value=15> 15<br><BR>
<input type=checkbox name="news" value=1> <B>Receber
Newsletter?</B><br><BR>
<input type=submit>
</form>
```

checkbox.php

```
<?php
// Verifica se usuário escolheu algum número
if(isset($_POST["numeros"]))
{
	echo "Os números de sua preferência são:<BR>";

	// Faz loop pelo array dos numeros
	foreach($_POST["numeros"] as $numero)
	{
		echo "- " . $numero . "<BR>";
	}
}
else
{
	echo "Você não escolheu número preferido!<br>";
}

// Verifica se usuário quer receber newsletter
if(isset($_POST["news"]))
{
	echo "Você deseja receber as novidades por email!";
}
else
{
	echo "Você não quer receber novidades por email...";
}
?>
```

6 - Campos Select

Os campos select permitem tratar uma variedade de opções, onde o usuário pode selecionar apenas uma opção ou múltiplas opções. Quando você permite múltiplas seleções, deve adicionar "[]" no final do nome, para o PHP interpretar como array.

Nos exemplos, mostro o funcionamento e tratamento de ambas.

Códigos Exemplos:

select.html

```
<form action="select.php" method="post">
<B>Qual seu processador?</B><br>
<select name=processador>
<option value=Pentium>Pentium</option>
<option value=AMD>AMD</option>
<option value=Celeron>Celeron</option>
</select><BR><BR>
<B>Livros que deseja comprar?</B><br>
Obs: segure "CTRL" para selecionar mais de um.<BR>
<select name="livros[]" multiple>
<option value="Biblia do PHP 4">Biblia do PHP 4</option>
```

```
<option value="PHP Professional">PHP Professional</option>
<option value="Iniciando em PHP">Iniciando em PHP</option>
<option value="Novidades do PHP 5">Novidades do PHP 5</option>
<option value="Biblia do MySQL">Biblia do MySQL</option>
</select><BR><BR>
<input type=submit>
</form>
```

select.php

```
<?php
echo "Seu processador é: " . $_POST["processador"] . "<BR>";

// Verifica se usuário escolheu algum livro
if(isset($_POST["livros"]))
{
	echo "O(s) livro(s) que você deseja comprar:<br>";
	// Faz loop para os livros
	foreach($_POST["livros"] as $livro)
	{
		echo "- " . $livro . "<br>";
	}
}
else
{
	echo "Você não escolheu nenhum livro!";
}
?>
```

Clique aqui para baixar os códigos desse artigo

Quaisquer dúvidas que tiver, não hesite em contatar-me!

Até a próxima semana, onde estarei mostrando como manipular o upload de arquivos de formulários e algumas boas técnicas para com formulários => PHP.

Alfred Reinold Baudisch
Desenvolvedor de Sistemas Web
alfred@auriumsoft.com.br
www.estacaonet.com

Excluir Marcados

```
<?php
/*
Banco - excluir_varios
Tabela
CREATE TABLE `produtos` (
  `id` int(11) NOT NULL,
```

```
  `produto` char(45) default NULL,
  `categoria` char(45) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
*/

// Form com arquivo chamando a si mesmo
$conexao = mysql_connect('localhost','root','ribafs');
mysql_select_db('excluir_varios',$conexao);
$consultar = "SELECT * FROM produtos ORDER BY id";
$resultado = mysql_query($consultar, $conexao);

if(mysql_num_rows($resultado) != 0){
    echo "<form name='frmExcluir' method='post' action='teste.php'>";
    echo "<table border=1><tr><th> </th><th>Produto:</th><th>Categoria:</th></tr>";

    while($linha = mysql_fetch_row($resultado)){
        echo "<td><input type='checkbox' name='id[]' value='$linha[0]'></td>
            <td>$linha[1]</td>
            <td>$linha[2]</td></tr>";
    }
    echo "<tr><td colspan='3'><input type='submit' name='excluir' value='Excluir!'></td></tr>";
    echo "</table></form>";
}else{
    echo "Nenhum registro foi encontrado!";
}
if(isset($_POST['id'])){
    $opcoes = $_POST['id'];
    $opcoes_text = implode(", ", $opcoes);
    $strexcluir = "DELETE FROM produtos WHERE id in (" . $opcoes_text . ")";
    mysql_query($strexcluir, $conexao) or die("Ocorreu algum erro");
}else{
    echo "É necessário escolher quem será excluído<br>";
    echo "<a href='javascript: history.back();'>Voltar</a>";
}
?>
```

Básico de GET e POST

Usar GET em situações seguras e POST em situações que precisem de segurança.

GET suporta apenas 1024 caracteres na string.
POST é sem limite.

Exemplo de uso de formulário com PHP chamando outro script

```
<body onLoad="document.form1.nome.focus()">
<form name="form1" method="POST" action="inserir.php">
Nome.<input type="text" name="nome"><br>
E-mail<input type="text" name="email"><br>
<input type="hidden" name="oculto" value="OK">
Senha<input type="password" name="senha"><br>
SenhaCript<input type="password" name="senhacript"><br>
<input type="submit" value="Enviar"><br>
</form>

<?php
//$nome=$_POST["nome"];
//$email=$_POST["email"];
if (isset($nome)){
    echo "O nome digitado foi " . $nome . "<br>";
    echo "O e-mail digitado foi " . $email . "<br>";
    echo "O campo oculto contém " . $oculto . "<br>";
    echo "A senha digitada foi " . $senha . "<br>";
    echo "A senha md5 digitada foi " . md5($senhacript) . "<br>";
}
?>
```

Exemplo de uso de formulário com PHP chamando o mesmo script

```
<body onLoad="document.form1.nome.focus()">
<form name="form1" method="POST" action="<?php $_PHPSELF ?>">
Nome.<input type="text" name="nome"><br>
E-mail<input type="text" name="email"><br>
<input type="submit" name="enviar" value="Enviar"><br>
</form>

<?php
if(isset($_POST['enviar'])){
$nome = $_POST["nome"];
$email = $_POST["email"];

    if(isset($nome)){
        echo "O nome digitado foi " . $nome . "<br>";
        echo "O e-mail digitado foi " . $email . "<br>";
    }
}
?>
```

import_request_variables("gP");
import_request_variables -- Import GET/POST/Cookie variables into the global scope

Muito indicado para quem tem o register_globals desativado e não quer digitar
$_POST, $_GET ou $_COOKIE.

Sempre deve vir na primeira linha do script ou antes do uso da variável.

```
import_request_variables("gP", "rvar_");

echo $rvar_foo;
```

Alerta: evite usar o recurso acima em produção, use apenas em caso de teste e ainda assim deve ser evitado e em teste devemos reproduzir o que iremos usar em produção.

As informações digitadas em um formulário podem ser capturadas por um script PHP.
Veja como:

form.html
```
<form action="recebe.php" method="POST">
Nome <input type="text" name="nome"><br>
Idade <input type="text" name="idade"><br>
<input type="submit" value="Enviar">
</form>
```

recebe.php

```
<?php
echo $_POST["nome"];
echo "<br>";
echo $_POST["idade"];
?>
```

Recebendo dados via URL

Se temos a seguinte situação:
Temos um arquivo c:\www\teste.php, com o seguinte código:

```
<?php
   echo $_GET['codigo'];
?>
```

Queremos passar uma variável código com valor 5, fazemos:
http://127.0.0.1/teste.php?codigo=5

Para receber mais de uma variável ou campo usar:
http://127.0.0.1/teste.php?codigo=5&nome="Antônio"&email="ribafs@yahoo.com"

```
echo "Código = ".$_GET['codigo']." Nome = ".$_GET['nome']." E-mail = ".$_GET['email'];
```

Também podemos passar variáveis pela URL, assim teste.php?codigo=$cod

9.3 - Trabalhando com Arquivos em PHP

Quando abrimos um arquivo para leitura (r+) o ponteiro situa-se no início do arquivo.
Quando abrimos um arquivo para anexar (a+) o ponteiro situa-se no final do arquivo.

Sempre usar fclose() quando finalizar de trabalhar com o arquivo para economizar memória.

O PHP suporta duas funções básicas para escrever em arquivos:
file_put_contents() - que escreve ou anexa uma string para um arquivo.
fwrite() - função que incrementalmente escreve dados para um arquivo texto.

Exemplos:

```
file_put_contents($string, $arquivo);
```

A função file($arquivo) lê arquivos texto.

Ler todo um arquivo em uma string:

```
$file = file_get_contents("/windows/system32/drivers/etc/services");
print("Size of the file: ".strlen($file)."\n");
```

Abrir um arquivo para leitura

```
$file = fopen("/windows/system32/drivers/etc/hosts", "r");
print("Tipo de manipulador de arquivo: " . gettype($file) . "\n");
print("Primeira linha do manipulador do arquivo: " . fgets($file));
fclose($file);
```

Ler um único caractere de um arquivo:

```
<?php
$file = fopen("/windows/system32/drivers/etc/services", "r");
$count = 0;
while ( ($char=fgetc($file)) !== false ) {
  if ($char=="/") $count++;
}
fclose($file);
print("Number of /: $count\n");
?>
```

Conteúdo do meuarquivo.txt

```
1|Meu nome|Meu site|Olá pessoal
2|Meu nome1|Meu site|Olá pessoal
3|Meu nome2|Meu site|Olá pessoal
4|Meu nome3|Meu site|Olá pessoal
5|Meu nome4|Meu site|Olá pessoal
```

Primeiro, vamos colocar todo o arquivo num array. Use o seguinte código:

```
$meuArray = file("nomedoarquivo.txt");
```

Vamos excluir a linha que tenha o primeiro valor do array igual a 3.

Para isto, temos que ler cada linha do array e fazer um "explode" delas, separando os "|" como se fosse em colunas.

Depois, faremos um [cfIF[/cf] perguntando se a coluna desejada é igual ao valor que queremos: 3

```
for($n=0; $n < count($meuArray); $n++) {
     $cadaLinha = explode("|", $meuArray[$n]);
    if($cadaLinha[0] == 3) {
        echo "Meu nome: $cadalinha[1] - Site: $cadalinha[2]";
    }
}
```

Excluindo linhas do array e salvando nos arquivos:

```
$meuArray = file("nomedoarquivo.txt"); // coloco todo o arquivo num array
$arrayModificado = array(); // crio um array vazio.

for($n=0; $n < count($meuArray); $n++) {
    $cadaLinha = explode("|", $meuArray[$n]);

    if($cadaLinha[0] <> 3) {
        $arrayModificado[] = $meuArray[$n];
    }
}
```

Depois é só salvar o array no arquivo, incluindo cada linha do array já modificado:

```
$bufferArquivo = fopen('nomedoarquivo.txt',"w");

for($n=0; $n < count($arrayModificado); $n++) {
   fwrite($bufferArquivo, $arrayModificado[$n]);
}

fclose($bufferArquivo);
```

Autor: Emannuel em http://www.codigofonte.net/dicas/php/130_trabalhando-com-arquivos

Veja também:
http://onlamp.com/pub/a/php/2002/10/03/php_foundations.html
http://onjava.com/pub/a/php/2002/12/12/php_foundations.html
http://onlamp.com/pub/a/php/2003/01/09/php_foundations.html

- 1 Abrir arquivo
- 2 Gravar em Arquivo
- 3 Ler Arquivo
- 4 fgets()
- 5 fgetc()
- 6 Ler Todo um Arquivo
- 7 file()
- 8 Ler Arquivo Via URL
- 9 feof(handle)
- 10 Contando o Número de Linhas de um arquivo
- 11 Contar palavras de um arquivo mostrando duplicadas
- 12 Ler de forma inversa um arquivo, linha a linha
- 13 Ler aleatoriamente linha de arquivo
- 14 Ler linha específica de arquivo
- 15 Operações com Diretórios
- 16 Mostrando conteúdo de diretório
- 17 Excluindo arquivos do SO
- 18 Copiando arquivos
- 19 Processando todos os arquivos de um diretório
- 20 Teste se Arquivo pode ser lido
- 21 Testar se Arquivo Permite Escrita
- 22 Testar se Arquivo Existe
- 23 Testar se é Arquivo ou Diretório
- 24 Outras Funções
- 25 Espaço Total no Disco
- 26 Espaço Livre no Disco
- 27 Tamanho de Diretório, número de arquivos e sub-diretórios
    - 27.1 Detalhes : $path
- 28 Tamanho de diretório
- 29 Trechos do Tutorial - The right way to read files with PHP da IBM
- 30 Lê e imprime todo o conteúdo de um arquivo CSV
- 31 rmdir -- Remove um diretório
- 32 rename -- Renomear um arquivo ($antigo, $novo)
- 33 unlink -- Apaga um arquivo
- 34 mkdir -- Criar um diretório
- 35 file_exists -- Checa se um arquivo ou diretório existe
- 36 disk_free_space -- Retorna o espaço disponivel no diretório
- 37 Receber Conteúdo de URL
- 38 Recursively find files by filename pattern
- 39 Referência

Abrir arquivo

fopen(filename, mode, [use_include_path]);

filename: pode ser simplesmente um nome, ou um caminho completo.

Exemplos: "arquivo.txt", "./arquivo.dat", "/data/data.txt".

mode: especifica o modo de abertura, ou seja, se o arquivo deve ser aberto para leitura, escrita, etc.
Modos de abertura:

```
- r: abre o arquivo no modo somente leitura e posiciona o ponteiro no início do
arquivo;
        o arquivo já deve existir;
- r+: abre o arquivo para leitura/escrita, posiciona o ponteiro no início do
arquivo;
- w: abre o arquivo no modo somente escrita; se o arquivo já existir, será
sobrescrito;
        senão, será criado um novo;
- w+: abre o arquivo para escrita/leitura; se o arquivo já existir, será
sobrescrito;
        senão, será criado um novo;
- a: abre o arquivo para anexar dados, posiciona o ponteiro no final do arquivo;
        se o arquivo não existir, será criado um novo;
- a+: abre o arquivo para anexo/leitura, posiciona o ponteiro no final do
arquivo;
        se o arquivo não existir, será criado um novo;

<?php
        $fp = fopen("./arquivo.dat", "r"); // $fp conterá o handle do arquivo
que abrimos
?>
<pre>


<h2>Fechar Arquivo</h2>

<pre>
fclose(handle_arquivo);

<?php
fclose($fp);
?>
```

Gravar em Arquivo

```
<?php
$fp = fopen("./dados.txt", "w");
fwrite($fp, "Olá mundo do PHP!"); // grava a string no arquivo. Se o arquivo não
existir ele será criado
fclose($fp);
?>
```

Ler Arquivo

```
<?php

$fp = fopen("./dados.txt", "r");
$texto = fread($fp, 20); // lê 20 bytes do arquivo e armazena em $texto
fclose($fp);
echo $texto;
?>
```

fgets()

Esta função é usada na leitura de strings de um arquivo. fgets() lê "length - 1" bytes do arquivo. Por default lê 1024 bytes para o comprimento da linha. Se for encontrado o final da linha e o número de bytes especificados não tiver sido atingido, fgets() terminará a leitura no final da linha (ou no final do arquivo, se for o caso). Eis a sua sintaxe:

fgets(handle, length);

- handle: handle do arquivo de onde os dados serão lidos;

- length: tamanho em bytes do buffer de leitura;

Exemplo:

```
<?php
$fp = fopen("./dados.txt", "r");
$texto = fgets($fp, 3);
fclose($fp);
echo $texto;
?>
```

fgetc()

Esta função permite ler caractere por caractere de um arquivo. Seguem a sintaxe e um exemplo de utilização:

fgetc(handle);

- handle: manipulador do arquivo de onde os dados serão lidos;

Ler Todo um Arquivo

```
<?php
$fp = fopen("./dados.txt", "r");
while (!feof($fp)){
    $char .= fgetc($fp);
}
fclose($fp);
echo $char."<br><br>";
?>
```

file()

Esta função lê um arquivo completo, e armazena cada linha do arquivo como um elemento de um array. Depois de ler todo o conteúdo do arquivo, file() o fecha automaticamente, não sendo necessária uma chamada a fclose(); Vejamos a sintaxe:

Ler Arquivo Via URL

$fh = fopen("http://127.0.0.1/", "r");

file(filename);

- filename: nome ou caminho completo de um arquivo.

Exemplo:

```
<?php
// file() lê todo o arquivo
$file_lines = file("./dados.txt");
echo "Primeira linha: " . $file_lines[0]."<br>";
echo "Segunda linha: " . $file_lines[1]."<br>";
echo "Terceira linha: " . $file_lines[2];
?>
```

Além dessas funções para leitura e escrita, existe ainda uma função bastante útil, que testa se o final do arquivo foi atingido. É a função feof(), que tem a seguinte sintaxe:

feof(handle)

- handle: handle do arquivo;

Exemplo:

```
<?php
$fp = fopen("./dados.txt", "r");
while(!feof($fp)) {
    $char .= fgetc($fp);
}
fclose($fp);
echo $char;
?>
```

Contando o Número de Linhas de um arquivo

```
<?php
// Contar o número de linhas de um arquivo, iniciando com 1
$fp = "./dados.txt";
$line_count = count (file ($fp));
echo $line_count;
?>
```

Contar palavras de um arquivo mostrando duplicadas

```
/*
// Contar palavras repetidas em um arquivo
$fn = "./dados.txt";
$f_contents = preg_split ("/[\s+/", implode ("", file ($fn)));
foreach ($f_content as $palavra) {
        $ar[$palavra]++;
}
print "A seguinte palavra tem duplicatas<br>";
foreach ($ar as $palavra => $conta_palavra) {
        if (conta_palavra > 1) {
                print "Palavra: $palavra<br>Número de ocorrências:
$conta_palavra<br><br>";
        }
}
*/
```

Ler de forma inversa um arquivo, linha a linha

```
<?php
$fn = "./dados.txt";
$f_contents = array_reverse (file ($fn));
foreach ($f_contents as $linha_inversa) {
        print $linha_inversa;
}
?>
```

Ler aleatoriamente linha de arquivo

```
<?php
$fn = "./pensamentos.txt";
$f_contents = file ($fn);
srand ((double)microtime()*1000000);
$linha_aleatoria = $f_contents[ rand (0, (count ($f_contents) - 1)) ];
print $linha_aleatoria;
?>
```

Ler linha específica de arquivo

```
<?php
$fn = "./dados.txt";
$nr_linha = 38;
$f_contents = file ($fn);
$sua_linha = $f_contents [$nr_linha];
print $sua_linha;
?>
```

Operações com Diretórios

Mostrando conteúdo de diretório

```
<?php
$dn = opendir ("/home/1www/");
while ($file = readdir ($dn)) {
        print "$file<br>";
}
closedir($dn);
?>
```

Excluindo arquivos do SO

```
<?php
$fn = "./dados0.txt";
// Excluindo arquivo
$ret = unlink ($fn);
if ($ret){
         die ("Arquivo excluído!");
}else{
        die ("Erro ao excluir arquivo");
}
?>
```

Copiando arquivos

```
<?php
$fn = "./dados.txt";

if (copy ($fn, "dados0.txt")){
         die ("Arquivo '$fn' copiado para dados0.txt ");
}else{
        die ("Erro ao copiar arquivo");
}
?>
```

Processando todos os arquivos de um diretório

```
<?php
$dh = dir ("/home/1www/");
while ($entrada = $dh->read()) {
        print $entrada . "<br>";
}
$dh->close();
?>
```

Teste se Arquivo pode ser lido

```
<?php
if (is_readable('http://127.0.0.1/index.html')) {
   header('Location: http://127.0.0.1/index.html');
}else{
        echo "Este arquivo não pode ser lido!";
}

?>


<?php
// TESTAR SE ARQUIVO PERMITE LEITURA

print '<br>';
$filename = 'teste2.php';
if (is_readable($filename)) {
   echo 'O arquivo permite leitura';
} else {
   echo 'O arquivo não permite leitura';
}
print '<br>';
```

//Outro

<?php if (is_readable('http://127.0.0.1/index.html')) {

```
  header('Location: http://127.0.0.1/index.html');
```

}else{ echo "Este arquivo não pode ser lido!"; }

?>

Testar se Arquivo Permite Escrita

```
if (is_writable($filename)) {
   echo 'O arquivo permite escrita';
} else {
   echo 'O arquivo não permite escrita';
}

?>
```

Testar se Arquivo Existe

```
<?php
print '<br>';
// TESTAR SE ARQUIVO EXISTE
$filename = 'teste2.php';

if (file_exists($filename)) {
```

```
   echo "O arquivo $filename existe";
} else {
   echo "O arquivo $filename não existe";
}
?>
```

Testar se é Arquivo ou Diretório

```
<?php
print '<br>';
// TESTAR SE ARQUIVO É UM ARQUIVO COMUN OU SE É DIRETÓRIO
$filename = 'teste2.php';

$filename2 = 'c:\windows';

if (is_file($filename)) {
   echo "O arquivo $filename é comun";
}else{
  echo "O arquivo $filename não é um arquivo comun";
}
print '<br>';

if (is_file($filename2)){
   echo "O arquivo $filename2 é comun";
}else{
  echo "O arquivo $filename2 não é um arquivo comun";
}

print '<br>';

if (is_dir($filename2)){
   echo "$filename2 é um diretório";
}else{
  echo "$filename2 não é um diretório";
}
```

Outras Funções

```
is_link($diretorio)
readlink($dir_link) // retorna o path completo do link
bool symlink ( string $destino, string $linkorigem ) // Cria um link simbólico
```

Espaço Total no Disco

<?php // ESPAÇO TOTAL NO DISCO $diretorio="c:/"; print disk_total_space($diretorio); print " ";

Espaço Livre no Disco

// ESPAÇO LIVRE NO DISCO // $df contém o número de bytes disponível em "/" $df = disk_free_space("c:/"); print $df ?>

Tamanho de Diretório, número de arquivos e sub-diretórios

<?php

// CALCULANDO TAMANHO OCUPADO POR UM DIRETÓRIO, NR DE ARQUIVOS E SUBDIRETÓRIOS

// http://www.go4expert.com/forums/showthread.php?t=290

function getDirectorySize($path) {

```
 $totalsize = 0;
 $totalcount = 0;
 $dircount = 0;
 if ($handle = opendir ($path))
 {
   while (false !== ($file = readdir($handle)))
   {
     $nextpath = $path . '/' . $file;
     if ($file != '.' && $file != '..' && !is_link ($nextpath))
     {
       if (is_dir ($nextpath))
       {
         $dircount++;
         $result = getDirectorySize($nextpath);
         $totalsize += $result['size'];
         $totalcount += $result['count'];
         $dircount += $result['dircount'];
       }
       elseif (is_file ($nextpath))
       {
         $totalsize += filesize ($nextpath);
         $totalcount++;
       }
     }
   }
 }
 closedir ($handle);
 $total['size'] = $totalsize;
 $total['count'] = $totalcount;
 $total['dircount'] = $dircount;
 return $total;
```

}

function sizeFormat($size) {

```
   if($size<1024)
   {
```

```
        return $size." bytes";
    }
    else if($size<(1024*1024))
    {
        $size=round($size/1024,1);
        return $size." KB";
    }
    else if($size<(1024*1024*1024))
    {
        $size=round($size/(1024*1024),1);
        return $size." MB";
    }
    else
    {
        $size=round($size/(1024*1024*1024),1);
        return $size." GB";
    }

}
```

// Usando

$path="D:/_xampplite/htdocs/desweb/7AplicativosExemplo/extras"; $ar=getDirectorySize($path);

echo "

Detalhes : $path

";

echo "Tamanho total : ".sizeFormat($ar['size'])."
"; echo "No. de arquivos : ".$ar['count']."
"; echo "No. de diretórios : ".$ar['dircount']."
"; ?>

Tamanho de diretório

<? // CALCULANDO TAMANHO (BYTES) OCUPADO POR UM DIRETÓRIO // http://www.weberdev.com/get_example-4171.html

function dir_size($dir) {

```
    if( !$dir or !is_dir( $dir ) )
    {
        return 0;
    }

    $ret = 0;
    $sub = opendir( $dir );
    while( $file = readdir( $sub ) )
    {
        if( is_dir( $dir . '/' . $file ) && $file !== ".." && $file !== "." )
        {
            $ret += dir_size( $dir . '/' . $file );
            unset( $file );
```

```
        }
        elseif( !is_dir( $dir . '/' . $file ) )
        {
            $stats = stat( $dir . '/' . $file );
            $ret += $stats['size'];
            unset( $file );
        }
    }
    closedir( $sub );
    unset( $sub );
    return $ret;

}
echo dir_size("D:/_xampplite/htdocs/desweb/1LinguagemPHP/php/tutoriais/trabalhando_com"); ?>
```

<?php

// MAIS UMA ÓTIMA FUNÇÃO PARA LER O TAMANHO DE UM DIRETÓRIO

```
/*
* PHP Freaks Code Library
* http://www.phpfreaks.com/quickcode.php
*
* Title: Directory Size
* Version: 1.0
* Author: Nathan Taylor aka(Lakario)
* Date: Saturday, 12/20/2003 - 12:34 PM
*
*
*
* NOTICE: This code is available from PHPFreaks.com code Library.
*         This code is not Copyrighted by PHP Freaks.
*
*         PHP Freaks does not claim authorship of this code.
*
*         This code was submitted to our website by a user.
*
*         The user may or may not claim authorship of this code.
*
*         If there are any questions about the origin of this code,
*         please contact the person who submitted it, not PHPFreaks.com!
*
*         USE THIS CODE AT YOUR OWN RISK! NO GUARANTEES ARE GIVEN!
*
* SHAMELESS PLUG: Need WebHosting? Checkout WebHost Freaks:
*                 http://www.webhostfreaks.com
*                 WebHosting by PHP Freaks / The Web Freaks!
```

- /
  // * Description / Example: // * // * This code will allow an individual to quickly obtain the size and number of files inside a directory recursively. // * // * It also includes a convenient byte value converter to kilobyte, megabyte, gigabyte, or trilobyte accordingly.

?>

Trechos do Tutorial - The right way to read files with PHP da IBM

USANDO fscanf

fscanf

Coming back to string processing, fscanf again follows the traditional C file library functions. If you're unfamiliar with it, fscanf reads field data into variables from a file.

```
list ($field1, $field2, $field3) = fscanf($fh, "%s %s %s");
```

FUNÇÃO fpassthru

No matter how you've been reading your file, you can dump the rest of your data to your standard output channel using fpassthru.

```
fpassthru($fh);
```

```
my_file = file_get_contents("myfilename");
echo $my_file;
```

Although it isn't best practice, you can write this command even more concisely as:

```
echo file_get_contents("myfilename");
```

This article is primarily about dealing with local files, but it's worth noting that you can grab, echo, and parse other Web pages with these functions, as well.

```
echo file_get_contents("http://127.0.0.1/");
```

This command is effectively the same as:

```
$fh = fopen("http://127.0.0.1/", "r");
fpassthru($fh);
```

You must be looking at this and thinking, "That's still way too much effort." The PHP developers agree with you. So you can shorten the above command to:

```
readfile("http://127.0.0.1/");
```

The readfile function dumps the entire contents of a file or Web page to the default output buffer. By default, this command prints an error message if it fails. To avoid this behavior (if you want to), try:

```
@readfile("http://127.0.0.1/");
```

Of course, if you actually want to parse your files, the single string that file_get_contents returns might be a bit overwhelming. Your first inclination might be to break it up a little bit with the split()

function.

$array = split("\n", file_get_contents("myfile"));

But why go through all that trouble when there's a perfectly good function to do it for you? PHP's file() function does this in one step: It returns an array of strings broken up by lines.

$array = file("myfile");

It should be noted that there is a slight difference between the above two examples. While the split command drops the newlines, the newlines are still attached to the strings in the array when using the file command (as with the fgets command).

PHP's power goes far beyond this, though. You can parse entire PHP-style .ini files in a single command using parse_ini_file. The parse_ini_file command accepts files similar to Listing 4.

Listing 4. A sample .ini file

```
; Comment
[personal information]
name = "King Arthur"
quest = To seek the holy grail
favorite color = Blue

[more stuff]
Samuel Clemens = Mark Twain
Caryn Johnson = Whoopi Goldberg
```

The following commands would dump this file into an array, then print that array:

```
$file_array = parse_ini_file("holy_grail.ini");
print_r $file_array;
```

The following output is the result:

```
Listing 5. Output
Array
(
    [name] => King Arthur
    [quest] => To seek the Holy Grail
    [favorite color] => Blue
    [Samuel Clemens] => Mark Twain
    [Caryn Johnson] => Whoopi Goldberg
)
```

Of course, you might notice that this command merged the sections. This is the default behavior, but you can fix it easily by passing a second argument to parse_ini_file: process_sections, which is a Boolean variable. Set process_sections to True.

$file_array = parse_ini_file("holy_grail.ini", true);

print_r $file_array;

And you'll get the following output:

Listing 6. Output

```
Array
(
    [personal information] => Array
        (
            [name] => King Arthur
            [quest] => To seek the Holy Grail
            [favorite color] => Blue
        )

    [more stuff] => Array
        (
            [Samuel Clemens] => Mark Twain
            [Caryn Johnson] => Whoopi Goldberg
        )

)
```

PHP placed the data into an easily parsable multidimensional array.

This is just the tip of the iceberg when it comes to PHP file processing. More complex functions like tidy_parse_file and xml_parse can help you handle HTML and XML documents, respectively. See Resources for details on how these particular functions work. These are well worth looking at if you'll be dealing with those types of files, but instead of considering every possible file type you might run into in detail in this article, here are a few good general rules for dealing with the functions I've described thus far.

Good practice

Never assume that everything in your program will work as planned. For example, what if the file you're looking for has moved? What if the permissions have been altered and you're unable to read the contents? You can check for these things in advance by using file_exists and is_readable.

Listing 7. Use file_exists and is_readable

```
$filename = "myfile";
if (file_exists($filename) && is_readable ($filename)) {
        $fh = fopen($filename, "r");
        # Processing
        fclose($fh);
}

In practice, however, such code is probably overkill. Processing the return
value of fopen is simpler and more accurate.

if ($fh = fopen($filename, "r")) {
        # Processing
        fclose($fh);
```

```
}
```

Final do trecho do tut IBM----------

Lê e imprime todo o conteúdo de um arquivo CSV

```
<?php
// Lê e imprime todo o conteúdo de um arquivo CSV
$row = 1;
$handle = fopen ("test.csv","r");
while ($data = fgetcsv ($handle, 1000, ",")) {
   $num = count ($data);
   print "<p> $num campos na linha $row: <br>";
   $row++;
   for ($c=0; $c < $num; $c++) {
       print $data[$c] . "<br>";
   }
}
fclose ($handle);
?>
<pre>

Outra:
<pre>
<?php

define('CSV_BOTH', 1);
define('CSV_ASSOC', 2);
define('CSV_NUM', 3);

function parse_csv($filename, $result_type = CSV_BOTH) {
   if(!file_exists($filename)) {
       die("file (" . $filename . ") does not exist\n");
   }

   $lines = file($filename);

   $title_line = trim(array_shift($lines));
   $titles = split(",", $title_line);

   $records = array();
   foreach($lines as $line_num => $line) {
       $subject = trim($line);
       $fields = array();
       for($field_num = 0; $field_num < count($titles); $field_num++) {
           if($subject{0} == '"') {
               preg_match('/^"(([^"]|\\")*)",?(.*)$/', $subject, $matches);

               $value = $matches[1];
               $subject = $matches[3];

               if($result_type == CSV_BOTH || $result_type == CSV_ASSOC) {
                   $fields[$titles[$field_num]] = $value;
               }
```

```
                if($result_type == CSV_BOTH || $result_type == CSV_NUM) {
                    $fields[$field_num] = $value;
                }
            } else {
                preg_match('/^([^,]*),?(.*)$/', $subject, $matches);

                $value = $matches[1];
                $subject = $matches[2];

                if($result_type == CSV_BOTH || $result_type == CSV_ASSOC) {
                    $fields[$titles[$field_num]] = $value;
                }

                if($result_type == CSV_BOTH || $result_type == CSV_NUM) {
                    $fields[$field_num] = $value;
                }
            }
        }

        $records[] = $fields;
    }

    return $records;
}

?>
<pre>

This version is conditional - it only adds quotes if needed:
<pre>
<?
function csv_escape($str) {
    $str = str_replace(array('"', ',', "\n", "\r"), array('""', ',', "\n", "\r"),
$str, &$count);
    if($count) {
        return '"' . $str . '"';
    } else {
        return $str;
    }
}
?>



<?php
$caminho = "/home/httpd/html/index.php";
$arquivo = basename ($caminho);        // $arquivo = "index.php"
$arquivo = basename ($caminho,".php"); // $arquivo = "index"
?>
```

rmdir -- Remove um diretório

This functions deletes or empties the directory. Without using recursive functions!

```
<?php
```

```
/**
 * Removes the directory and all its contents.
 *
 * @param string the directory name to remove
 * @param boolean whether to just empty the given directory, without deleting
the given directory.
 * @return boolean True/False whether the directory was deleted.
 */
function deleteDirectory($dirname,$only_empty=false) {
   if (!is_dir($dirname))
       return false;
   $dscan = array(realpath($dirname));
   $darr = array();
   while (!empty($dscan)) {
       $dcur = array_pop($dscan);
       $darr[] = $dcur;
       if ($d=opendir($dcur)) {
           while ($f=readdir($d)) {
               if ($f=='.' || $f=='..')
                   continue;
               $f=$dcur.'/'.$f;
               if (is_dir($f))
                   $dscan[] = $f;
               else
                   unlink($f);
           }
           closedir($d);
       }
   }
   $i_until = ($only_empty)? 1 : 0;
   for ($i=count($darr)-1; $i>=$i_until; $i--) {
       echo "\nDeleting '".$darr[$i]."' ... ";
       if (rmdir($darr[$i]))
           echo "ok";
       else
           echo "FAIL";
   }
   return (($only_empty)? (count(scandir)<=2) : (!is_dir($dirname)));
}

?>

//Outra

<?php
/* Function to remove directories, even if they contain files or
subdirectories.  Returns array of removed/deleted items, or false if nothing
was removed/deleted.

by Justin Frim.  2007-01-18

Feel free to use this in your own code.
*/

function rmdirtree($dirname) {
   if (is_dir($dirname)) {    //Operate on dirs only
```

```
        $result=array();
        if (substr($dirname,-1)!='/') {$dirname.='/';}    //Append slash if
necessary
        $handle = opendir($dirname);
        while (false !== ($file = readdir($handle))) {
            if ($file!='.' && $file!= '..') {    //Ignore . and ..
                $path = $dirname.$file;
                if (is_dir($path)) {    //Recurse if subdir, Delete if file
                    $result=array_merge($result,rmdirtree($path));
                }else{
                    unlink($path);
                    $result[].=$path;
                }
            }
        }
        closedir($handle);
        rmdir($dirname);    //Remove dir
        $result[].=$dirname;
        return $result;    //Return array of deleted items
    }else{
        return false;    //Return false if attempting to operate on a file
    }
}
?>
```

rename -- Renomear um arquivo ($antigo, $novo)

```
<?php
  rename("/tmp/tmp_file.txt", "/home/user/login/docs/my_file.txt");
?>
```

unlink -- Apaga um arquivo

mkdir -- Criar um diretório

mkdir ("/path/to/my/dir", 0700);

is_file -- Diz se o arquivo é um arquivo comum (não é diretório)

file -- Le um arquivo inteiro para um array

```
<?php
// Le um arquivo em um array. Nesse exemplo você pode obter via HTTP para obter
// o código fonte HTML de uma URL.
$lines = file ('http://www.exemplo.com/');

// Roda através do array, mostrando o fonte HTML com numeração de linhas.
foreach ($lines as $line_num => $line) {
   echo "Linha #<b>{$line_num}</b> : " . htmlspecialchars($line) . "<br>\n";
}
// Outro exemplo, onde obtemos a página web inteira como uma string. Veja também
file_get_contents().
$html = implode ('', file ('http://www.exemplo.com/'));
?>
```

file_exists -- Checa se um arquivo ou diretório existe

```
$filename = '/caminho/para/qualquer.txt';

if (file_exists($filename)) {
   print "O arquivo $filename existe";
} else {
   print "O arquivo $filename não existe";
}
```

disk_free_space -- Retorna o espaço disponivel no diretório

Receber Conteúdo de URL

<?php // Trazer conteúdo de arquivo ou de página para string

// Define a context for HTTP. $aContext = array(

```
  'http' => array(
      'proxy' => 'tcp://10.0.0.1:3128', // This needs to be the server and the
port of the NTLM Authentication Proxy Server.
      'request_fulluri' => True,
      ),
  );
```

$cxContext = stream_context_create($aContext);

// Now all file stream functions can use this context. $sFile = file_get_contents("http://www.google.com", False, $cxContext); echo $sFile; ?>

Recursively find files by filename pattern

http://snippets.dzone.com/posts/show/4147

Scans a directory, and all subdirectories for files, matching a regular expression. Each match is sent to the callback provided as third argument. A simple example:

function my_handler($filename) {

```
 echo $filename . "\n";
```

} find_files('c:/', '/php$/', 'my_handler');

And the actual snippet

function find_files($path, $pattern, $callback) {

```
 $path = rtrim(str_replace("\\", "/", $path), '/') . '/';
```

```
 $matches = Array();
 $entries = Array();
 $dir = dir($path);
 while (false !== ($entry = $dir->read())) {
   $entries[] = $entry;
 }
 $dir->close();
 foreach ($entries as $entry) {
   $fullname = $path . $entry;
   if ($entry != '.' && $entry != '..' && is_dir($fullname)) {
     find_files($fullname, $pattern, $callback);
   } else if (is_file($fullname) && preg_match($pattern, $entry)) {
     call_user_func($callback, $fullname);
   }
 }

}


<?php

$file='file.inc';
$php_errormsg = 'Erro na operação do arquivo!';

// Determinaod onde o penteiro está
/*
  $fr = fopen($file, 'r');
  fseek($fr, rand(1, 1024));
  $offset = ftell($fr);
  echo "We randomly landed on the $offset byte of the file.<BR>";
*/
?>

<?php
// Escreve no início do arquivo
$file_name= $file;
if(file_exists($file_name))
{
    //open file for writng and place pointer at the end
    $handle = fopen($file_name, 'a+');

    if(!$handle)
    {
        die("couldn't open file <i>$file_name</i>");
    }

    //place pointer at the beginning of the file.
    //rewind($handle);

    $data = fgets($handle, 4096);
```

```
    // move de volta para o inicio do arquivo
    // o mesmo que rewind($fp);
    fseek($handle, 0);

    //write to file
    //fwrite($handle, "Student ID: 12345");
    echo "success writing to start of file";
} else {
    echo "file <i>$file_name</i> doesn't exists";
}

fclose($handle);

// Posição atual
$fp = fopen($file, 'a+');
$data = fgets($fp, 4096);
echo '<br>Onde estamos '.ftell($fp).'<br>';
fclose($fp);


// Retornar a última linha do arquivo

function readlastline($file)
{
      $fp = @fopen($file, "r");
      $pos = -1;
      $t = " ";
      while ($t != "\n") {
           fseek($fp, $pos, SEEK_END);
           $t = fgetc($fp);
           $pos = $pos - 1;
      }
      $t = fgets($fp);
      fclose($fp);
      return $t;
}
print readlastline($file);

?>

<?php
/*
$file_name= $file;
if(file_exists($file_name))
{
    //open file for writng and place pointer at the end
    $handle = fopen($file_name, 'a+');
```

```
    if(!$handle)
    {
        die("couldn't open file <i>$file_name</i>");
    }

    fwrite($handle, "Student GPA: 2.9");
    echo "success writing to file";
} else {
    echo "file <i>$file_name</i> doesn't exists";
}

fclose($handle);
*/
?>


<?php

/*
// move de volta para o inicio do arquivo
// o mesmo que rewind($fp);
fseek($fp, 0);
// Escrevendo com quebra de linha
$file_name = "myfile.txt";
if(file_exists($file_name))
{
    //open file for writng and place pointer at the end
    $handle = fopen($file_name, 'w');

    if(!$handle)
    {
        die("couldn't open file <i>$file_name</i>");
    }
    $str.= "Student Name: Mark Fendisen\r\n";
    $str.= "Student ID: 12345\r\n";
    $str.= "Student GPA: 2.9\r\n";

    fwrite($handle, $str);
    echo "success writing to file";
} else {
    echo "file <i>$file_name</i> doesn't exists";
}

fclose($handle);
*/
?>
```

Referência
http://phpbrasil.com/articles/print.php/id/310

9.4 - Trabalhando com Diretórios no PHP

```
$Dir = “C:\\PHP”;
$DirOpen = opendir($Dir);
while ($CurFile = readdir($DirOpen)) {
        echo $CurFile . “<br />”;
}
closedir($DirOpen);

$Dir = “C:\\PHP”;
$DirEntries = scandir($Dir);
foreach ($DirEntries as $Entry) {
        echo $Entry . “<br />”;
}

mkdir($diretorio);

is_writable($dir);

file_exists($dir);

is_dir($dir);

copy($arq_origem, $arq_destino);

unlink($arquivo);

rmdir($dir);



<?php
// Listar conteúdo de diretório
function listar_dir($arquivo){
    $dn = opendir ($arquivo);
    while ($file = readdir ($dn)) {
     return "$file<br>";
    }
    closedir($dn);
}

// Espaço total em disco em KB
function total_space($dir){
    return round(disk_total_space($dir)/(1024*1024),2) . ' KB';
}
//print total_space('/home/ribafs');

// Espaço livre em disco em KB
```

```
function free_space($dir){
    return round(disk_free_space($dir)/(1024*1024),2) . ' KB';
}
//print free_space('/home/ribafs');

// Receber tamanho de diretório recursivamente
function getdirsize($dir) {
    // Initialize the size value
    $size = 0;
    // Get the files and directories
    // inside the given directory
    $files = scandir($dir);
    // Loop through the files
    foreach ($files as $file) {
        // Filter-out .. and .
        if (strlen(str_replace('.', '', $file))>0) {
            if (is_dir($dir.'/'.$file)) {
                // If the item is a directory
                // run this function again and
                // get the size of the files inside
                $size += getdirsize($dir.'/'.$file);
            } else {
                // If it's a file, calculate the size
                // and add it to the size variable
                $size += filesize($dir.'/'.$file);
            }
        }
    }
    // Finally, return the calculated size
    return $size;
}
//print getdirsize('/home/ribafs/uteis');

function dir_size($directory, $type, $round) {
  $dir = scandir($directory);
  $size = 0;
  foreach($dir as $value) {
    if(is_dir($directory.'/'.$value)) {
      $init = scandir($directory.'/'.$value);
        foreach($init as $subfile) {
          $filesize = filesize($directory.'/'.$value.'/'.$subfile);
          $size = $size + $filesize;
        }
      if(is_file($directory.'/'.$value)) {
        $filesize = filesize($directory.'/'.$value);
        $size = $size = $filesize;
      }
    }
```

```
  }
  if($type==1) {
    // returns KB
    $size = $size / 1024;
    if(isset($round))
      $size = round($size, $round);
    echo $size.'KB';
  }
  elseif($type==2) {
    $size = $size / 1048576;
    if(isset($round))
      $size = round($size, $round);
    echo $size.'MB';
  }
  elseif($type==3) {
    $size = $size / 1073741824;
    if(isset($round))
      $size = round($size, $round);
    echo $size.'GB';
  }
  else {
    if(isset($round))
      $size = round($size, $round);
    echo $size.'Bytes';
  }
}
/*
 * #1 with how you would like the data to be shown (1 = KB, 2 = MB, 3 = GB, 0 or nothing is
Bytes), and #2 is optional, if you want it to make the number perttier enter how many numbers
AFTER the . to have, so if you put in 2, and the number before it rounded it was 3.4325214 it would
be something like, 3.44 or so after it rounds it.
 */
dir_size("/home/ribafs", 1, 2);


function dir_size2( $dir )
{
    if( !$dir or !is_dir( $dir ) )
    {
        return 0;
    }

    $ret = 0;
    $sub = opendir( $dir );
    while( $file = readdir( $sub ) )
    {
        if( is_dir( $dir . '/' . $file ) && $file !== ".." && $file !== "." )
        {
```

```
            $ret += dir_size( $dir . '/' . $file );
            unset( $file );
        }
        elseif( !is_dir( $dir . '/' . $file ) )
        {
            $stats = stat( $dir . '/' . $file );
            $ret += $stats['size'];
            unset( $file );
        }
    }
    closedir( $sub );
    unset( $sub );
    return $ret;
}

//print dir_size2('/home/ribafs/htdocs');

/**
 * Removes the directory and all its contents.
 *
 * @param string the directory name to remove
 * @param boolean whether to just empty the given directory, without deleting the given directory.
 * @return boolean True/False whether the directory was deleted.
 */
function delete_dir($dirname,$only_empty=false) {
  if (!is_dir($dirname))
      return false;
  $dscan = array(realpath($dirname));
  $darr = array();
  while (!empty($dscan)) {
      $dcur = array_pop($dscan);
      $darr[] = $dcur;
      if ($d=opendir($dcur)) {
          while ($f=readdir($d)) {
              if ($f=='.' || $f=='..')
                  continue;
              $f=$dcur.'/'.$f;
              if (is_dir($f))
                  $dscan[] = $f;
              else
                  unlink($f);
          }
          closedir($d);
      }
  }
  $i_until = ($only_empty)? 1 : 0;
  for ($i=count($darr)-1; $i>=$i_until; $i--) {
      echo "\nDeleting '".$darr[$i]."' ... ";
```

```
    if (rmdir($darr[$i]))
        echo "ok";
    else
        echo "FAIL";
  }
  return (($only_empty)? (count(scandir)<=2) : (!is_dir($dirname)));
}

/* Function to remove directories, even if they contain files or
subdirectories.  Returns array of removed/deleted items, or false if nothing
was removed/deleted.

by Justin Frim.  2007-01-18

Feel free to use this in your own code.
*/

function rmdirtree($dirname) {
  if (is_dir($dirname)) {    //Operate on dirs only
    $result=array();
    if (substr($dirname,-1)!='/') {$dirname.='/';}    //Append slash if necessary
    $handle = opendir($dirname);
    while (false !== ($file = readdir($handle))) {
        if ($file!='.' && $file!= '..') {    //Ignore . and ..
            $path = $dirname.$file;
            if (is_dir($path)) {    //Recurse if subdir, Delete if file
                $result=array_merge($result,rmdirtree($path));
            }else{
                unlink($path);
                $result[].=$path;
            }
        }
    }
    closedir($handle);
    rmdir($dirname);    //Remove dir
    $result[].=$dirname;
    return $result;    //Return array of deleted items
  }else{
    return false;    //Return false if attempting to operate on a file
  }
}

?>
```

9.5 - Trabalhando com Path no PHP

- 1 PATH
- 2 Exemplos simples de uso de funções para path do PHP
- 3 Recebendo o Path Absoluto do Script Atual
- 4 Recebendo o path relativo do webserver do script atual

Exemplos simples de uso de funções para path do PHP

```
<?php
   $path=dirname(realpath($_SERVER['SCRIPT_FILENAME']));
   $path=substr($path,0,strlen($path) - 5);
echo "<br>Path deste script sem 5 finais caracteres - " . $path;

echo "<br><br>Diretório atual - ".dirname(__FILE__);
echo "<br>Caminho completo do script atual - ".__FILE__;

echo "<br>URL do script atual - " . "http://" . $_SERVER['HTTP_HOST'] .
$HTTP_SERVER_VARS["SCRIPT_NAME"];
?>

<?php
$path = "/etc/passwd";
$file = dirname($path); // $file is set to "/etc"
?>

<?php
$path = "/home/httpd/html/index.php";
$file = basename($path);        // $file is set to "index.php"
$file = basename($path, ".php"); // $file is set to "index"
?>

<?php
echo dirname($_SERVER["REQUEST_URI"]);
?>
```

Recebendo o Path Absoluto do Script Atual

dirname(__FILE__)

Recebendo o path relativo do webserver do script atual

```
function GetRelativePath($path){
   $npath = str_replace('\\', '/', $path);
   return str_replace(GetVar('DOCUMENT_ROOT'), '', $npath);
}
```

GetRelativePath(dirname(__FILE__));

<?php

if (DIRECTORY_SEPARATOR=='/')

```
 $absolute_path = dirname(__FILE__).'/';
```

else

```
 $absolute_path = str_replace('\\\\', '/', dirname(__FILE__)).'/';
```

?>

Resultará em um path absoluto no estilo UNIX que funciona também em PHP5 sob Windows.

Em algumas instalações (< 4.4.1) $_SERVER['REQUEST_URI'] não está configurado, usado o código para corrigir:

```
<?php

if (!isset($_SERVER['REQUEST_URI'])) {
   $_SERVER['REQUEST_URI'] = substr($_SERVER['PHP_SELF'],1);
   if (isset($_SERVER['QUERY_STRING'])) $_SERVER['REQUEST_URI'].='?'.
$_SERVER['QUERY_STRING'];
}

?>

$my_uri = "http://" . $_SERVER['HTTP_HOST'] . $HTTP_SERVER_VARS["SCRIPT_NAME"];

// então

<?php echo ("$my_uri");?>

<?php
include("{$_SERVER['DOCUMENT_ROOT']}/includes/my_include.php");
?>
```

Você pode usar isso para receber o diretório pai:

dirname(dirname(__FILE__))

...include a file relative to file path:

include(dirname(__FILE__) . '/path/relative/file_to_include.php');

Isso colocará ambos os paths "www" e "file" de forma fácil para transportar o array.

```
<?php
// build the www path:
$me = $_SERVER['PHP_SELF'];
$Apathweb = explode("/", $me);
$myFileName = array_pop($Apathweb);
$pathweb = implode("/", $Apathweb);
$myURL = "http://".$_SERVER['HTTP_HOST'].$pathweb."/".$myFileName;
$PAGE_BASE['www'] = $myURL;
```

```
// build the file path:
strstr( PHP_OS, "WIN") ? $strPathSeparator = "\\" : $strPathSeparator = "/";
$pathfile = getcwd ();
$PAGE_BASE['physical'] = $pathfile.$strPathSeparator.$myFileName;

// this is so you can verify the results:
$www = $PAGE_BASE['www'];
$physical = $PAGE_BASE['physical'];

echo "$physical<p>";
echo "$www<p>";
?>
```

retornará algo como:

Windows:

F:\dev\Inetpub\wwwroot\somedirectory\index.php

http://devserver/somedirectory/index.php

Unix:

/home/somepathto/gieson.com/webroot/index.php

http://www.gieson.com/index.php

Path absoluto do script em execução

$path=dirname(realpath($_SERVER['SCRIPT_FILENAME']));

```
print 'Diretório atual=> '.dirname($_SERVER["SCRIPT_FILENAME"]);
print '<br>Diretório atual=> '.dirname(__FILE__);
print '<br>Diretório anterior=> '.dirname(dirname(__FILE__));
print '<br>Diretório anterior 2 níveis=> '.dirname(dirname(dirname(__FILE__)));
print '<br>Diretório raiz geral => '.dirname($_SERVER['PHP_SELF']);
print '<br>Diretório atual isolado => '.basename(dirname(__FILE__));
print '<br>Arquivo atual isolado => '.$_SERVER['REQUEST_URI'];
print '<br>Arquivo atual path completo => '.__FILE__;
print '<br>Arquivo raiz geral => '.dirname($_SERVER["SCRIPT_NAME"]);
```

```
$site_path = realpath(dirname(__FILE__));
define ('__SITE_PATH', $site_path);
```

```
function fixPath ($path)
(
    return dirname($path . '/.');
)
```

```
fixPath('/one');            // '/one'
fixPath('/one/');           // '/one'
fixPath('/one//two');       // '/one/two'
fixPath('/one////two////'); // '/one/two'
```

Checar permissão de escrita:

```
error_reporting(E_ALL);
$dir_name = '/var/www/virtual/phpintra/htdocs/php/';
do {
   $b_is_writable = is_writable($dir_name);
   echo sprintf("Dir[%s]Writable[%s]\n", $dir_name, $b_is_writable? 'YES':'NO');
}while (($dir_name = dirname($dir_name)) !='/');

//Restringindo acesso direto
if (!defined("BASE_PATH")) define('BASE_PATH',
dirname($_SERVER['SCRIPT_NAME'])=='/' ? './' : str_repeat("../",
substr_count(dirname($_SERVER["SCRIPT_NAME"]), "/")));


function myRealPath($path) {
   // Check if path begins with "/". => use === with strpos
   if (strpos($path,"/") === 0) {
      $path = $_SERVER['DOCUMENT_ROOT'].$path;
   }
   else {
      // Strip slashes and convert to arrays.
      $currentDir = preg_split("/\/",dirname($_SERVER['PATH_TRANSLATED']));
      $newDir = preg_split('/\/',$path);
      // Drop one directory from the array for each ".."; add one otherwise.
      foreach ($newDir as $dir) {
         if ($dir == "..")
            array_pop($currentDir);
         elseif ($dir != ".") //test for "." which represents current dir (do nothing in that case)
            array_push($currentDir,$dir);
      }
      // Return the slashes.
      $path = implode($currentDir,"/");
   }
   return $path;
}
//print myRealPath("./");

$p = getcwd(); // recebe o diretório atual
//echo $p;

$dir_path = str_replace( $_SERVER['DOCUMENT_ROOT'], "", dirname(realpath(__FILE__)) ) .
```

```
DIRECTORY_SEPARATOR;
//print $dir_path;

/*
echo realpath(dirname(__FILE__)); //Similar ao getcwd()

$inc = $_SERVER["DOCUMENT_ROOT"]."/inc";
include $inc."/config.php";
include $about."/setup.php";
include $inc."/header.php";
include $inc."/topnav.php";
include $inc."/left.php";
include $inc."/right.php";


echo __FILE__AbsolutePath(__FILE__)."\n";

function __FILE__AbsolutePath($Filename)
 {
      switch (PHP_OS)
       {
            case "WINNT": $needle = "\\"; break;
            case "Linux": $needle = "/";  break;
            default:      $needle = "/";  break; // TODO : Mac check
       }
      $AbsPath = substr($Filename, 0, strrpos($Filename, $needle));
      return $AbsPath;
 }
*/
function __FILE__AbsolutePath()
 {
      $d = debug_backtrace();
      $Filename = $d[0]['file'];

      switch (PHP_OS)
       {
            case "WINNT": $needle = "\\"; break;
            case "Linux": $needle = "/";  break;
            default:      $needle = "/";  break; // TODO : Mac check
       }
      $AbsPath = substr($Filename, 0, strrpos($Filename, $needle));
      return $AbsPath;
 }
//print __FILE__AbsolutePath() ;
```

9.6 - Trabalhando com Includes em PHP

INCLUDE - inclue e avalia o conteúdo do arquivo.
REQUIRE - também inclue e avalia o conteúdo do arquivo incluído.

A diferença entre ambos é que o include ao encontrar um erro, lança um warning apenas, enquanto que o require lança um Fatal Error, que pára o processamento.

Uso do require: para códigos que requerem maior segurança.

INCLUDE_ONCE e REQUIRE_ONCE - são semelhantes ao include e require, sendo que estes incluem um arquivo somente uma vez.

Mostrando uso do require_once:

echo.php
```
<?php
echo "Hello";
?>
```

teste.php
```
<?php
require('echo.php');
require_once('echo.php');
?>
```

Executar teste.php
saída: "Hello".

Agora teste2.php:
```
<?php
require('echo.php');
require('echo.php');
?>
```

Executar teste2.php
saída: "HelloHello".

Agora teste3.php:
```
<?php
require_once('echo.php');
require('echo.php');
?>
```

Executar teste3.php
saída: "HelloHello".

Ou seja, ao encontrar require_once, ele verifica se o arquivo já foi incluído, e somente o incluirá

novamente se ele ainda não tiver sido incluído.

```
<?php
// Isto está errado e não funcionará como desejado
if ($condition)
  include $arquivo;
else
  include $outro;


// E este está correto
if ($condition) {
  include $arquivo;
} else {
  include $outro;
}

?>
```

```
<?php
$path="/full/path/to/script/";
if (getdomain($path) == 'yourdomain'){
    include($path.'somefile.php');
}
?>
```

"variables.php"

```
<?php
$includer = basename($_SERVER['SCRIPT_NAME']);

switch ($includer) {
  case 'a.php':
  $this_variable = 'included by script a.php';
  break;

  case 'b.php':
  $this_variable = 'included by script b.php';
  break;

  default:
  $this_variable = 'included by unkown script';
}
echo $this_variable;
?>
```

Test with 3 different files "a.php", "b.php", "c.php", all with the same content:

```
<?php
include 'variables.php';
?>
```

9.7 - Trabalhando com Funções no PHP

O PHP conta com uma grande quantidade de boas bibliotecas de funções internar sobre vários assuntos, como strings, arrays, arquivos, diretórios, data e hora, etc.

Além das funções embutidas também podemos criar nossas próprias funções quando não encontrarmos uma função que atenda as nossas necessidades.

Funções são blocos de código, conjuntos de instruções, que ficam quietos, sem vida, até que sejam chamadas em algum lugar do código. Funções reduzem o trabalho de digitação, o tamanho dos scripts em geral, os erros, reaproveitando código, facilitando e organizando o trabalho de desenvolvimento.

- 1 Exemplos de funções definidas pelo usuário
- 2 Variáveis globais
- 3 Variável externa acessível na função
- 4 Acessando variáveis Externas
    - 4.1 Variável externa inacessível
- 5 Variáveis estáticas
- 6 Recursiva
- 7 Declarando variáveis static
- 8 Retornar mais de um valor de uma função
- 9 Passando Argumentos através de Funções
- 10 Por Valor
- 11 Por Referência
- 12 Otimização do tempo de execução
- 13 Função com parâmetro default
- 14 Trabalhando com Funções OB

Exemplos de funções definidas pelo usuário

```
function quadrado($numero) {
        print $numero*$numero;
}
```

Executando:

quadrado(6); // Saída: 36

Variáveis globais

```
$var1 = 5;
function testeGlobal1(){
        $var1 = 1;
        print "<br>Valor de \$var1: $var1";
}
echo testeGlobal1();
```

Variável externa acessível na função

```
$var2 = 10;
function testeGlobal2(){
        global $var2;
        print "<br>Valor de \$var2 $var2";
}
echo testeGlobal2();

$var5 = 15;
function testeGlobal5(){
        $var5 = 5;
    print "<br><br>A variável global vale $GLOBALS[var5], ";
    print "Já a variável local vale $var5<br><br>";
}

testeGlobal5();

function cliente($codigo, $idade = 18){
        print "Código = $codigo, Idade = $idade";
}

cliente(1);  //Exibirá: Código = 1, Idade = 18

function cubo($num){
        return ($num*$num*$num);
}

$var1 = 2 * cubo(5);echo "<br>".$var1;
```

Acessando variáveis Externas

Normalmente de dentro de uma função não temos acesso às variáveis que estão fora dela. Mas veja como contornar isso:

Variável externa inacessível

```
$var1 = 5;
function testeGlobal1(){
        $var1 = 1;
        print ”Valor de \$var1: $var1”;
}
echo testeGlobal1();
Variável externa acessível na função

$var2 = 10;
function testeGlobal2(){
        global $var2;
        print ”Valor de \$var2 $var2”;
}
echo testeGlobal2();
```

Outra alternativa é usar o array $GLOBALS[], que contém todas as variáveis globais. Veja um exemplo:

```
$var5 = 15;
function testeGlobal5(){
        $var5 = 5;
        print "<br><br>A variável global vale $GLOBALS[var5], ";
        print "Já a variável local vale $var5<br><br>";
}

testeGlobal5();
```

Variáveis estáticas

```
function contador(){
        static $x = 0;
        return $x++;
}
echo "<br>";
echo contador();echo contador();echo contador();
//A saída será: 012

function contador2(){
        $x = 0;
        return $x++;
}
echo "<br>";
echo contador2();echo contador2();echo contador2();
//A saída será: 000.

  function staticfunction() {
   static $count = 0;
```

```
   $count++;
        if ($count==1){
       echo "A Função foi executada $count vez<br>";
        }else{
           echo "A Função foi executada $count vezes<br>";
        }
  }

  for($i = 0; $i < 5; $i++) {
   staticfunction();
  }

function Teste1()
{
    static $a = 0;
    echo $a;
    $a++;
}

for($x=0;$x<=10;$x++){
        echo Teste1()." ";
}
```

Recursiva

```
function Teste()
{
    static $count = 0;

    $count++;
    echo $count." ";
    if ($count < 10) {
        Teste ();
    }
    $count--;
}

Teste();
```

Declarando variáveis static

```
function foo(){
    static $int = 0;          // correro
   // static $int = 1+2;        // errado (é uma expressão)
   // static $int = sqrt(121);  // errado  (é uma expressão também)

    $int++;
    echo $int;
}
```

```
function aumentoSalario($sal, $perc=5){
        $salario = $sal * $perc/100;
        echo $salario;
}
echo "<br>Aumento: " . aumentoSalario(1956);

function redirecionar($url){
        header("Location: $url");
}
echo "<br>";
redirecionar("http://ribafs.phpnet.us/");
echo "<br>";
```

Retornar mais de um valor de uma função

usa-se arrays e list()

array() retorna e list() exibe

//Exemplo:

```
function recebe_hoje(){
        $data_abreviada=date("d/m/Y");
        $data_extensa=date("l, d F \d\e Y");

        return array($data_abreviada, $data_extensa);
}

list($data_abreviada, $data_extensa)=recebe_hoje();
        print $data_extensa;
echo "<br>";
        print $data_abreviada;


<h2>Declaração dinâmica de função</h2>

<pre>
if ($f == 1){
        function f1(){
                echo "funcao1";
        }
}else{
        function f2(){
                echo "funcao2";
        }
}

<h2>Retornando o número de argumentos de uma função</h2>

<pre>
function ret_args_funcao() {
    $numargs = func_num_args();
    echo "Número de argumentos: $numargs<br>\n";
    if ($numargs >= 2) {
    echo "Segundo argumento vale : " . func_get_arg (1) . "<br>\n";
    }
    $arg_list = func_get_args();
```

```
    for ($i = 0; $i < $numargs; $i++) {
    echo "Argumento $i vale: " . $arg_list[$i] . "<br>\n";
    }
}

ret_args_funcao (1, 2, 3);
```

Passando Argumentos através de Funções

O default é 'por valor', que passa uma cópia do valor da variável.

Também podemos passar 'por referência', onde se passa o endereço da própria variável. Quando atribuímos uma variável a outra passando como referência, não fazemos cópia alguma, mas sim passamos o endereço da variável original, portanto qualquer alteração nesta referência refletirá na variável original.

ByRef é mais eficiente em termos de memória ao lidar com variáveis e arrays grandes e também permite alterar o valor da variável original, o que não acontece com o ByVal, mas a vantagem de desempnho somente é percebida em grandes arrays em grandes loops.

Para maior segurança setar:

allow_call_time_pass_reference no php.ini

Impede a passagem de valores por referência nas chamadas, mas permite somente na definição das funções.

$var1 = & $var2;

Agora ambas apontam para o mesmo endereço e valor.

Reter valor de variáveis entre chamadas (static)

(Guarda o valor da última chamada até o final da execução do script, tantas vezes quantas a função for chamada).

Exemplo:

```
$valor = 4;
$ref = &$valor;

$ref = 3;

$valor = 4;
$ref = &$valor; // Aqui tornamos ambas as variáveis com o mesmo endereço
                // O que alterarmos em $ref alteramos em $valor

$ref = 3;       // Com isso também alteramos $valor para 3, veja abaixo.
echo $valor . "<br>";

$valor=0;          // Com isso também alteramos $ref para 0, veja abaixo.

echo $ref;
```

Por Valor

```
function val_subtracao($num1, $num2){
        if($num1 < $num2){
                die("Números negativos");
        }else{
                $return_result=0;
                while($num1 > $num2){
                        $num1 = $num1 - 1;
                        $return_result = $return_result + 1;
                }
        }
        return($return_result);
}

$primeiro_op=493;
$segundo_op=355;
$resultado1 = val_subtracao($primeiro_op, $segundo_op);
print ("Resultado1 é $resultado1<br>");
$resultado2 = val_subtracao($primeiro_op, $segundo_op);
print("Resultado2 é $resultado2<br>");
```

Por Referência

```
function subtracao_ref(&$num1, &$num2){
        if($num1 < $num2){
                die("Números negativos");
        }else{
                $return_result=0;
                while($num1 > $num2){
                        $num1 = $num1 - 1;
                        $return_result = $return_result + 1;
                }
        }
        return($return_result);
}

$primeiro_op=493;
$segundo_op=355;
$resultado1 = subtracao_ref($primeiro_op, $segundo_op);
print ("<br><br>Resultado1 é $resultado1<br>");
$resultado2 = subtracao_ref($primeiro_op, $segundo_op);
print("Resultado2 é $resultado2<br>");

echo "Agora, se se nós executarmos exatamente a mesma chamada da subtração como
fizemos a primeira vez, receberemos a saída:
resultado1 é 138 e resultado2 é 0";


/*
Sugestão de chamada de função:

if (nome_funcao($argumento){
        ....
        ....
```

```
}else{
        ....
}

*/
?>

<?php
// Retorna o tipo e o valor de variável
function ss_as_string (&$thing, $column = 0) {
    if (is_object($thing)) {
        return ss_object_as_string($thing, $column);
    }
    elseif (is_array($thing)) {
        return ss_array_as_string($thing, $column);
    }
    elseif (is_double($thing)) {
        return "Double(".$thing.")";
    }
    elseif (is_long($thing)) {
        return "Long(".$thing.")";
    }
    elseif (is_string($thing)) {
        return "String(".$thing.")";
    }
    else {
        return "Unknown(".$thing.")";
    }
}

// Retorna o tipo e o valor de array
function ss_array_as_string (&$array, $column = 0) {
    $str = "Array(<BR>\n";
    while(list($var, $val) = each($array)){
        for ($i = 0; $i < $column+1; $i++){
            $str .= "    ";
        }
        $str .= $var.' ==> ';
        $str .= ss_as_string($val, $column+1)."<BR>\n";
    }
    for ($i = 0; $i < $column; $i++){
        $str .= "    ";
    }
    return $str.')';
}

// Retorna o tipo e o valor de objeto
function ss_object_as_string (&$object, $column = 0) {
    if (empty($object->classname)) {
        return "$object";
    }
    else {
        $str = $object->classname."(<BR>\n";
        while (list(,$var) = each($object->persistent_slots)) {
            for ($i = 0; $i < $column; $i++){
                $str .= "    ";
            }
```

```
            global $$var;
            $str .= $var.' ==> ';
            $str .= ss_as_string($$var, column+1)."<BR>\n";
        }
        for ($i = 0; $i < $column; $i++){
            $str .= "    ";
        }
        return $str.')';
    }
}

$var="Riba";
echo ss_as_string($var);
//echo ss_as_string($GLOBALS);
```

Otimização do tempo de execução

```
function ss_timing_start ($name = 'default') {
    global $ss_timing_start_times;
    $ss_timing_start_times[$name] = explode(' ', microtime());
}

function ss_timing_stop ($name = 'default') {
    global $ss_timing_stop_times;
    $ss_timing_stop_times[$name] = explode(' ', microtime());
}

function ss_timing_current ($name = 'default') {
    global $ss_timing_start_times, $ss_timing_stop_times;
    if (!isset($ss_timing_start_times[$name])) {
        return 0;
    }
    if (!isset($ss_timing_stop_times[$name])) {
        $stop_time = explode(' ', microtime());
    }
    else {
        $stop_time = $ss_timing_stop_times[$name];
    }
    // do the big numbers first so the small ones aren't lost
    $current = $stop_time[1] - $ss_timing_start_times[$name][1];
    $current += $stop_time[0] - $ss_timing_start_times[$name][0];
    return $current;
}

?>
```

Função com parâmetro default

Parâmetro default é aquele cujo valor já faz parte da função, sendo opcional.

```
function cliente($codigo, $idade = 18){
        print "Código = $codigo, Idade = $idade";
}
```

```
cliente(1);  //Exibirá: Código = 1, Idade = 18
```

Trabalhando com Funções OB

- Artigo do Adriano Oliveira Gonçalves na Revista do PHP

9.8 - Trabalhando com Session no PHP

Sessões são usadas para guardar dados enquanto a janela do browser estiver aberta. São geralmente usadas para manter dados como nome do usuário, tipo do usuário (se é o administrador ou se é um visitante não cadastrado, por exemplo), entre outros dados importantes. Dica: NUNCA COLOQUE A SENHA NA SESSÃO. VOCÊ SÓ PRECISA DELA PARA AUTENTICAÇÃO!
Para iniciar uma sessão no PHP usamos a função:

```
session_start();
```

Se quisermos destruir a sessão (no caso de o usuário ter feito logoff), usamos a função:

```
session_destroy();
```

Agora precisamos apenas setar os valores que ficarão na sessão. No PHP os valores são armazenados em um vetor associativo chamado $_SESSION. As associações são feitas com pares chave e valor. Por exemplo, vamos setar o nome do usuário e a permissão dele, e colocar esses dados na seção. Vamos buscar esses dados a partir de um formulário fictício, via POST.

```
< ?php
    session_start(); //iniciamos a sessão
    $_SESSION['usuario'] = $_POST['usuario']; //colocamos na sessão o valor do
campo usuário vindo do formulário
    $_SESSION['permissao'] = $_POST['admin']; //da mesma forma setamos suas
permissões
    header("location:pagina_principal.php"); //Redirecionamos para a pagina
principal
? >
```

Com isso feito, podemos acessar as variáveis da sessão de qualquer lugar da nossa aplicação através do vetor $_SESSION.

Fonte: http://maozinhadaweb.blogspot.com/2007/05/tutorial-de-php-parte-3-sesses-e.html

Sessions em PHP

Session ou sessão é o método ou via pela qual podemos gravar informação de variáveis, funcionando assim como uma variável global. A grande vantagem das sessions é podermos navegar por várias páginas sem que esta informação se perca evitando assim passar o valor das variáveis indivídualmente de página em página.

Ao contrário dos cookies que ficam gravadas no computador do utilizador, as sessions são gravadas no servidor e automaticamente eliminadas quando o utilizador fecha o browser.

Exemplificando:
Vamos criar a página pagina1.php

```
<?php
//Iniciar sessão
session_start();
//Gravar variáveis em sessão
$_SESSION['id']=1;
$_SESSION['nome']='Paulo';
//Imprimir o valor das variáveis guardadas
print("$id");
print("$nome");
?>
```

Podemos verificar que são imprimidos os valores das variáveis atribuídas à session: 1 e Paulo, ou seja, esses valores ficam gravados numa session e podem ser utilizados em qualquer página onde essa sessão esteja aberta.

Modificar e remover sessions

Quando utilizamos sessions podemos ter a necessidade de modificar ou remover variáveis de sessão.
Para alterar o valor de uma variável basta fazer nova atribuição:

```
$_SESSION['nome']='Pedro';
```

A variável $nome passaria a ter o valor de Pedro a partir desta linha.

Para remover todas as variáveis da session basta colocar o comando session_unset() ou apenas unset() para remover variáveis indivídualmente.
Para eliminar ou destruir completamente toda a sessão utilizamos o comando session_destroy().

Como referido anteriormente, a sessão acaba sempre que o utilizador fechar o browser, mas a duração das sessões também podem ser alteradas para um espaço de tempo pre-definido. Para isso deverá ser alterado o php.ini na linha session.cookie_lifetime = 0 colocando o número de segundos desejados no lugar do 0 (zero).

Gravar sessions num directório

As sessões também podem ser gravadas num directório no servidor. Para que esta operação seja realizada devera ser dada a instrução através da função session_save_path().
Nesta função deverá ser dado o caminho para a pasta que guardará as sessões:
session_save_path("sessions"). Neste caso a pasta sessions que guardará as sessões fica localizada na raís.

Para que esta função funcione deverá ser colocada sempre antes do comando session_start().

Vamos experimentar:

```
 <?php
//Pasta onde vai gravar  sessão
```

```
session_save_path("sessions");
//Iniciar sessão
session_start();
//Gravar variáveis em sessão
$_SESSION['id']=1;
$_SESSION['nome']='Paulo';
//Imprimir o valor das variáveis guardadas
print("$id");
print("$nome");
?>
```

Pode confirmar que é gravado um ficheiro na pasta sessions com um nome do tipo "sess_t4k884rd8uunp8ojt83picc9j1". Se abrir este ficheiro com o Notepad pode verificar que se encontram lá gravadas as variáveis dadas à sessão.

Convém salientar que, por razões óbvias, a pasta onde são gravadas as sessions deverá estar protegida.

Fonte: http://www.techcaffe.net/site/tutoriais.php?recID=4

Exemplo prático de Session em PHP

index.php

```
<?php
session_start();
?>
<h1 align="center">Usando SESSION em PHP</h1>
Podemos preservar valores de variáveis enquanto durar uma sessão do browser através do uso de SESSION.<br>
Para isso devemos startar a sessão em cada página que desejamos usar esta variável com<br><br>
sesssion_start();<br><br>
Lembrando que esta função deve vir antes de qualquer comando que mande algo para a tela, caso<br>
o session esteja configurado para usar cookie.<br>
Na primeira página deve ter um formulário com algum campo que devemos usar no session.<br>
Experimente gravar a URL de uma das páginas internas e acessar diretamente (http://localhost/session3)
<br>Primeiro feche todas as seções do browser e depois abra o browser com essa URL.<br>
<br>
Veja que SESSION é muito bom para preservar o valor de variáveis entre páginas de um site numa seção.<br>
```

Portanto seu uso é muito útil quando pretendemos autenticas os visitantes de todas as as páginas de um site.

Como também para outros usos em que pretendemos reaproveitar o valor de variáveis (algo como global).

Acompanhe este exemplo para ver detalhes.


```
<form method=post action=session2.php>
Login<input type=text size=8 name=login><br>
<input type=submit value=Enviar>
</form></center>
```

session2.php

```
<?php
session_start();
$_SESSION['login']=$_POST['login'];
if (isset($_SESSION['login'])){
echo "Entre. Session2. <a href=session3.php>Session3</a>";
} else {
echo "Acesso não autenticado!";
}
?>
```

session3.php

```
<?php
session_start();
if (isset($_SESSION['login'])){
  echo "Entre. Session3. <a href=session4.php>Session4</a><br>";
  echo "<a href=destruir.php>Drestruir Sessão</a>";
} else {
  echo "Acesso não autenticado!";
}
?>
```

session4.php

```
<?php
session_start();
if (isset($_SESSION['login'])){
echo "Entre. Session4. <a href=session5.php>Session5</a>";
} else {
echo "Acesso não autenticado!";
}
?>
```

session5.php

```
<?php
session_start();
if (isset($_SESSION['login'])){
echo "Entre. Session5. <a href=index.php>Index</a><br><br>";
echo "Informações: <br>ID da Sessão: <b>" . session_id() . "</b><br>Variável mantida pela SuperGlobal \$_SESSION: <b>" . $_SESSION['login'];
} else {
echo "Acesso não autenticado!";
}
?>
```

Dicas sobre Session:

```
print ini_get('max_execution_time');

ini_set("session.gc_maxlifetime","3600");
session_save_path("/my/own/path/without/url/sessions");
```

Bom tutorial em: http://www.phpro.org/tutorials/Introduction-To-PHP-Sessions.html

9.9 - Trabalhando com Cookies no PHP

Setando o Cookie

Para gravar um cookie no navegador do visitante usa-se a função setcookie:

setcookie*(nome, valor, data_expiração)*:

Exemplo:

```
//Calculate 60 dias no futuro
//segundos * minutos * horas * dias + tempo atual
$emDoisMeses = 60 * 60 * 24 * 60 + time();
setcookie('UltimaVisita', date("G:i - d/m/Y"), $emDoisMeses);
```

Recebendo o Cookie

```
if(isset($_COOKIE['UltimaVisita']))
   $visita = $_COOKIE['UltimaVisita'];
else
   print "You've got some stale cookies!";

echo "Sua última visita foi - ". $visita;
```

Fonte: http://www.tizag.com/phpT/phpcookies.php

9.10 - Tratamento de Erros no PHP

Os 25 erros de programação mais perigosos segundo a SANS
By coredump

Saiu no site da SANS a lista criada com o consenso entre varios profissionais e empresas do ramo de segurança e desenvolvimento descrevendo os 25 erros de programação mais perigosos para o desenvolvimento seguro. Eu vou traduzir os nomes e informação básicos mas o melhor é ler o artigo na íntegra, em inglês.

Os erros estão separados em três categorias: Interação insegura entre componentes, Gerenciamento arriscado de recursos, Defensas porosas.

Categoria: Interação insegura entre componentes

1. **Validação Imprópria de Entradas**: Entradas que recebem dados e os aceitam mesmo sem certificar que eles são do tipo/formato esperado.
2. **Codificação ou Escape Impróprios de Saída**: Saídas que não são codificadas ou escapadas corretamente são a maior fonte de ataques de injeção de código.
3. **Falha ao Preservar a Estrutura da Busca SQL (conhecido como Injeção de SQL)**: Se os atacantes podem influenciar as procuras SQL do seu programa, então eles podem controlar o seu banco de dados.
4. **Falha ao Preservar a Estrutura do Código da Página (conhecido como "Cross-site Scripting")**: Assim como o anterior, se os atacantes podem injetar código ou scripts em sua página, eles podem controlar a página.
5. **Falha ao Preservar a Estrutura de Comandos do Sistema Operacional**: Se você permitir que entradas ilegais sejam passadas para aplicativos do sistema operacional, o atacante pode controlar o servidor.
6. **Transmissão de Dados Sensíveis em Texto Puro**: Senhas, dados de cartão e qualquer informação considerada sensível deve ser criptografada.
7. **Falsificação de Requisição Entre Sites**: Um atacante pode criar uma requisição que é enviada a outro site forjando a origem e fazendo o mesmo partir de um usuário inocente, aproveitando credenciais de autenticação e acessos.
8. **Condição de Corrida**: Atacantes vão sempre procurar por condições de corrida no software para conferir se alguma informação importante não é obtida no processo.
9. **Vazamento de Informações em Mensagens de Erro**: Atacantes vão procurar por mensagens de erro que descrevam mais que o necessário, como nomes de campos SQL, objetos e bibliotecas sendo utilizadas.

Categoria: Gerenciamento arriscado de recursos:

1. **Falha ao Limitar Operações aos Limites de um Buffer de Memória**: O conhecido buffer overflow.
2. **Controle Externo de Dados Sensíveis**: Informações críticas que são mantidas fora de um banco de dados por questões de performance não deviam ser facilmente acessíveis por atacantes.
3. **Controle Externo de de Caminho ou Nome de Arquivo**: Quando você usa dados externos para montar um nome de arquivo ou caminho de gravação, você está se arriscando a ser atacado.
4. **Caminho de Procura Inseguro**: Se o caminho de procura de recursos estiver em algum lugar sob controle de um atacante, bibliotecas ou código pode ser inserido a revelia.

5. **Falha ao Controlar a Geração de Código**: Caso o atacante consiga influenciar a geração de código dinâmico (se geração de código dinâmico for utilizada no programa) ele poderá controlar todo seu código.
6. **Download de Código sem Verificação de Integridade**: Se você executa código obtido por download, você confia na fonte. Atacantes podem aproveitar esta confiança.
7. **Desligamento ou Liberação Impróprias de Recursos**: Arquivos, conexões e classes precisam ser corretamente encerradas.
8. **Inicialização Imprópria**: Dados, bibliotecas e sistemas inicializados incorretamente podem abrir margens para problemas.
9. **Cálculos Incorretos**: Quando o atacante tem algum controle sobre as entradas usadas em operações matemáticas, isso pode gerar vulnerabilidades.

Categoria: Defensas porosas:

1. **Controle de Acesso Impróprio**: Se você não garante que seus usuários estão fazendo apenas o que deviam, os atacantes irão se aproveitar de sua autenticação.
2. **Uso de um Algoritmo Criptográfico Quebrado ou Vulnerável**: Utilização de algoritmos fracos ou comprometidos levam a falhas de criptografia e vulnerabilidades.
3. **Senha no Código**: deixar um usuário e uma senha no próprio código traz inúmeros problemas.
4. **Permissão de Acesso Insegura para Recurso Crítico**: Configurações, arquivos de dados e bancos de dados devem ter suas permissões de acesso protegidas.
5. **Uso de Valores Insuficientemente Aleatórios**: Se você usa tipos de segurança que dependem de aleatoriedade, usar um gerador aleatório insuficiente só vai causar problemas.
6. **Execução com Privilégios Desnecessários**: Se seu programa precisa de privilégios elevados para executar suas funções, ele deve abrir mão destes direitos assim que ele termina de executar as ações que precisavam dos privilégios.
7. **Aplicação de Segurança do Lado do Servidor pelo Cliente**: Atacantes podem usar engenharia reversa em um cliente de software e escrever seus próprios clientes removendo testes e aplicações de segurança.

Algumas coisas foram realmente chatas de traduzir, sinta-se livre para sugerir correções.

intel

PS: Lembre-se sempre “os 25 mais” não quer dizer “os 25 únicos”. *Grain of Salt* faz bem.

Fonte: http://core.eti.br/2009/01/22/os-25-erros-de-programacao-mais-perigosos-segundo-a-sans/

Exemplo simples de uso de Exception

```
<?php
/*
http://www.sourcecode4you.com/article/articleview/956dd1bb-a5a2-42ad-ae31-ad836eec24f3/try-catch-block-in-php.aspx

Try Catch Block In Php

Try catch block handle runtime exception which occur in function.
*/

function dividebyZero()
{
    try
    {
        $k= 10/0;
    }
    catch(Exception $ex)
    {
        echo $ex;
    }
}

dividebyZero();

function throwException()
{
    try
    {
        throw new Exception("My Custom Exception");
    }
    catch(Exception $ex)
    {
        echo $ex;
    }
}
throwException();
?>
```

Introdução a manipulação de erros em PHP

Autor: Lorran Luiz <lorranluiz at click21.com.br>
Data: 22/01/2009

Introdução

Na história humana muitos conquistaram grandes coisas. Tiveram problemas é claro, mas em suas decisões não erraram ou erraram pouco (pois é, com os erros que aprendemos). Os erros enfim tiveram um papel relativamente importante para o sucesso dessas pessoas. Foi com eles (os erros) que as soluções apareceram, pois não existe solução se não há um erro para ser consertado.

É claro que erros nos decepcionam, mas eles só refletem a realidade - de que ainda não estamos completamente qualificados a fazer tal ato com êxito completo. Os erros mostram que não somos perfeitos.

A fato de não sermos perfeitos esbarra no que fazemos. Enfim, os programadores erram, os usuários podem errar ou até mesmo alguma parte do programa que está a ser executado pode falhar de alguma maneira.

São nesses erros que envolvem o aplicativo e/ou os atores envolvidos no sistema que grandes desastres começam a se formar. Verdadeiras bolas de neve vão crescendo a medida que um programador digita seu código sem preocupar-se com a menor possibilidade de algo dar errado.

Depois que um sistema está completamente construído, sem base alguma para o tratamento correto de erros, o "problema quase que irresolvível" estará montado! A cada passo na vida desse sistema a possibilidade da ocorrência de erros "não identificáveis" aumentará, e a cada erro, mais e mais do seu tempo se consumirá para se chegar a resolução do mesmo. O sistema enfim se torna insustentável e todas aquelas horas de trabalho foram desperdiçadas!

Situações como estas poderiam ser evitadas se medidas simples como a construção de um código "supervisionado" com cláusulas e blocos que garantem a manipulação correta e segura de erros fossem implementadas logo nas primeiras linhas.

Situações de possíveis erros

Veremos a partir de então o básico quando o assunto é manipulação de erros. Com tal conhecimento o programador se capacitará a avançar no estudo de tratamento de erros e posteriormente aprender técnicas para tal.

Algumas situações

Suspeite de qualquer situação em que seu código fará algo decisivo para o funcionamento do seu sistema. Coisas como conectar a um banco de dados, abrir algum arquivo que contenha parâmetros importantes, carregar uma classe (ou uma biblioteca) essencial na aplicação ou armazenar alguma informação numa tabela de banco de dados, entre muitas outras situações requerem uma execução sem falhas e por isso deve haver uma preocupação maior com os erros que acontecerem.

Podemos dizer que em certas partes de seu código necessitaremos de "vigilantes" atentos a possíveis situações anormais. Veremos então quem são os vigilantes da programação no PHP.

Os "vigilantes"

Podemos rodear determinadas áreas de seu script com "vigilantes", que esperam que algo dê errado para passar o controle da situação aos seus "superiores".

Usamos a palavra-chave *try* para representar esses vigias virtuais no seu código fonte.

O try é seguido de um bloco de códigos no qual ele estará responsável por capturar eventuais erros. Os erros são denominados exceções, ou seja, situações que saem do esperado.

Sintaxe:

```
try {
//Código suspeito
}
```

Exceções

Em orientação a objeto, no PHP, exceções representam um objeto. Esse objeto necessita de um "molde" para ser construído, esse molde é denominado *Classe* e a classe usada para "modelar" uma exceção é a classe *Exception*.

Abaixo vemos a classe Exception:

```
<?php
class Exception {

  protected $message = 'Unknown exception'; // Mensagem da exceção
  protected $code = 0;                      // Código da exceção definido pelo usuário
  protected $file;                          // Arquivo gerador da exceção
  protected $line;                          // Linha geradora da exceção

  function __construct(string $message=NULL, int code=0);

  final function getMessage();              // Mensagem da exceção
  final function getCode();                 // Código da exceção
  final function getFile();                 // Arquivo gerador
  final function getTrace();                // um array com o backtrace()
  final function getTraceAsString();        // String formatada do trace

  /* Sobrecarregável */
  function _toString();                     // String formatada para ser mostrada

}
```

?>

A classe Exception não precisa ser inserida em seu código, ela é uma classe nativa do PHP introduzida no Zend 2.

Os "superiores"

Os "superiores" são aqueles que tem competência para fazer algo, devem resolver um problema ou passar essa responsabilidade para quem resolva. No PHP usamos a palavra-chave *catch* para representar esse superior.

O "catch" é responsável pela identificação do erro, ou seja, ele "sabe" qual o tipo de exceção ocorreu, de acordo com a classe, derivada de Exception, que "modelou" o objeto de exceção passado para ele.

O "catch" necessita de um parâmetro, o nome da classe Exception ou derivada seguida da variável que conterá o objeto da exceção e um bloco de código.

Sintaxe:

```
catch (Exception $variavel) {
//Código que manipulará tal exceção
}
```

A palavra-chave catch deve ser posicionada logo após o bloco de código rodeado pela cláusula try. Ficando então assim:

Sintaxe:

```
try { //Código suspeito } catch (Exception $variavel) { //Código que manipulará tal exceção }
```

Preparando seu código para as exceções

Já vimos como cercar uma parte do código para manipular os erros que nela ocorrerem. Veremos a partir de então como "delatar" os erros e alertar o sistema quanto a eles.

Construindo uma exceção

O método construtor da classe Exception requer 2 parâmetros, sendo 1 opcional:

```
public Exception::__construct ([ string $message=NULL [, int $code ]] )
```

O primeiro parâmetro é a mensagem do erro e o segundo o código do erro (opcional).

Disparando o alarme

Em PHP usamos a palavra-chave *throw* para alertar o sistema da ocorrência de um erro, uma

exceção. Throw é seguido de um código que solicita a construção de um novo objeto de exceção.

Para instanciar uma classe no PHP usamos a palavra-chave *new* seguida do nome da classe. Então a sintaxe ficará assim:

```
throw new Exception("Mensagem", 1);
```

Como usar "throw"

Podemos usar a condicional *if* para decidirmos quando disparar uma exceção.

Veja o exemplo abaixo:

```
if (!@mysql_connect ("localhost", "usuário", "senha")) throw new Exception("Não foi possível conectar ao banco de dados");
```

Obs.: Podemos usar o operador de supressão (@) de erros para evitar que sejam exibidas as mensagens de erro padrão do PHP.

Estendendo e especificando exceções

Podemos tornar nossas exceções mais específicas e por consequência saber com mais precisão que tipo de erro ocorreu, manipulando-o assim da melhor maneira possível.

Usamos a palavra-chave *extends* para estender uma classe, ou seja, criar uma classe derivada (ou classe filha). Quando uma classe estende outra, ela herda da última atributos e métodos dependendo do nível de acesso de cada elemento que constitui o corpo da classe. A classe filha pode sobrecarregar métodos (redefiní-los).

Exception é uma classe que pode ser herdada, sendo assim podemos definir classes que derivem e aumentem sua funcionalidade, passando de uma simples exceção genérica para exceções mais precisas.

Exceções derivadas

Definamos algumas classes derivadas de Exception:

```
class DBException extends Exception {
   protected $variavel; //Uma variavel qualquer

   public function __construct($exceptionMessage, $errorCode = 0) {
      parent::__construct($exceptionMessage, $errorCode);
   }

   //Aqui vemos alguns métodos característicos da classe derivada de Exception
   public function getVariavel() {
      return $this->variavel_qualquer;
   }
```

```
    public function setVariavel($valor = NULL) {
        $this->variavel = $valor;
    }
}

class FopenException extends Exception {
    protected $arquivo; //Variável contendo o nome de um arquivo

    public function __construct($nomeDoArquivo, $exceptionMessage, $errorCode = 0) {
        parent::__construct($exceptionMessage, $errorCode);
        $this->arquivo = $nomeDoArquivo;
    }

    //Métodos característicos dessa classe
    public function getNomeDoArquivo() {
        return $this->arquivo;
    }

    public function exibirNomeDoArquivo() {
        echo "Nome do arquivo: {$this->arquivo}";
    }

}
```

Um código básico com tratamento de erros

Veremos e entenderemos como funcionaria então um código básico completo que trataria de uma maneira certa os possíveis erros que ocorressem durante sua execução.

O código

O código que se seguirá é meramente explicativo e por isso talvez não possua utilidade numa situação real.

- Criaremos um código onde será criado um manipulador de arquivos;
- O conteúdo do arquivo será lido;
- Efetuaremos uma conexão com o servidor de banco de dados;
- Esse conteúdo será inserido numa célula de uma tabela de um banco de dados;
- E se tudo ocorrer bem, será exibida a mensagem: Operação efetuada com sucesso;
- Caso ocorra um erro será exibida uma mensagem de erro que será reportada ao programador.

Veja abaixo:

```
//A CLASSE:

class ArmazenadorDeArquivoEmDB {
```

```
   protected $db_obj;
   protected $db_usuario;
   protected $db_senha;
   protected $db_dsn;
   protected $nomeDoArquivo;
   protected $manipuladorDoArquivo;
   protected $conteudoDoArquivo;
   protected $db_tabela;
   protected $db_dbName;

   public function __construct($usuario, $senha, $tabela, $db, $servidor) { //Construir objeto
      $this->db_usuario = $usuario;
      $this->db_senha = $senha;
      $this->db_tabela = $tabela;
      $this->db_dbName = $db;
      $this->db_dsn = $servidor;
   }

   public function conectar() {
      if(!($this->db_obj = @mysql_connect($this->db_dsn, $this->db_usuario, $this->db_senha))) {
         throw new DBException("Não foi possível conectar ao banco de dados", 281); //Lembrando
que o código de erro não é um parâmetro obrigatório
      }
   }

   public function armazenarArquivoNoDB($nomeDoArquivo) { //O parâmetro requer endereço
completo do arquivo
      $this->nomeDoArquivo = $nomeDoArquivo;
      try {
            if (empty($this->db_obj)) { //Se não existir uma conexão aberta com o DB
               $this->conectar();
            }
         } catch  ( DBException $e ) { //Caso haja algum erro durante a tentativa de conexão...
            throw $e;                    // ...o "alarme" irá disparar
         }

      //Iremos abrir o arquivo:
      if (!file_exists($nomeDoArquivo)) throw new FopenException($nomeDoArquivo, "O arquivo
especificado não existe.", 282);
      if (!($this->manipuladorDoArquivo = @fopen($nomeDoArquivo, "r"))) throw new
FopenException($nomeDoArquivo, "Não foi possível abrir um manipulador para este arquivo.",
283);
      if (!($this->conteudoDoArquivo = @fread($this->manipuladorDoArquivo,
intval(sprintf("%u", filesize($nomeDoArquivo)))))) throw new FopenException($nomeDoArquivo,
"Não foi possível abrir o conteúdo deste arquivo.", 284); //O 2º parâmetro de fread() possibilita a
leitura de arquivos maiores que 2GB

      //Armazenaremos agora o conteúdo do arquivo no DB
```

```
        $sql = 'INSERT INTO `' . $this->db_dbName . '`.`' . $this->db_tabela . '` (`nome`, `conteudo`)
VALUES (\" . $this->nomeDoArquivo . '\', \". $this->conteudoDoArquivo . '\');';
        if(!($insert = @mysql_query($sql))) throw new DBException("Não foi possível inserir o
arquivo no banco de dados.", 285);
        else echo '<html><head></head><body onload="alert(\'O arquivo foi armazenado com
sucesso no banco de dados\')"></body></html>';
    }
}

//O OBJETO EM AÇÃO:

$armazenadorDeArquivoEmDB = new ArmazenadorDeArquivoEmDB("usuario", "senha",
"tabela", "banco_de_dados", "sevidor"); //Criamos um objeto que armazena um arquivo num banco
de dados

try { //Tentiva de conexão
    $armazenadorDeArquivoEmDB->armazenarArquivoNoDB("Ergue-te Marcos.txt"); //Aqui entra
o nome do arquivo a ser armazenado no DB
} catch ( DBException $e ) {
    //Houve um erro relativo ao banco de dados
    echo nl2br("<b>{$e->getMessage()}</b>\n<br />Detalhes:\n{$e->__toString()}"); //Exibir string
contendo informações sobre a exceção
    $gossipy->reportar($e); //A nível exemplificativo, temos este objeto fictício, imaginário, que
envia por email informações sobre o erro ao programador
} catch ( FopenException $e ) {
    //Houve um erro referente ao arquivo, e podemos dar um tratamento específico a este tipo de
exceção!
    echo nl2br("<b>{$e->getMessage()}</b>\n<br />Detalhes:\n{$e->__toString()}"); //Exibir string
contendo informações sobre a exceção
    $gossipy->reportar($e); //A nível exemplificativo, temos este objeto fictício, imaginário, que
envia por email informações sobre o erro ao programador
}
```

Poderíamos ter explorado mais desta fantástica linguagem que é o PHP e ter feito um tratamento de erros mais proveitoso no código acima, mas acho que para fins meramente explicativos e didáticos não é necessário tanta complexidade.

Vocês podem ver que poderíamos ter criado no código acima subclasses de Exception bem mais específicas, a idéia é a mesma de quando criamos as classes "FopenException" e "DBException". Classes mais específicas para a manipulação de exceções relativas ao banco de dados poderiam ser definidas apenas estendendo a classe "DBException", veja:

```
SQLException extends DBException { }
```

E cada vez podemos ir tornando as exceções mais específicas (e a manipulação mais precisa).

Resumindo

Faremos então uma compilação de tudo o que aprendemos:

- Neste artigo revisamos conceitos básicos do poderoso recurso que é a orientação a objetos do PHP 5;
- Vimos algumas situações de risco que podem surgir durante a execução de seu código;
- Vimos também que podemos aprender com as situações frustradoras se tratarmos melhor as informações relativas a aquele problema;
- Vimos que o PHP 5 conta com excelentes ferramentas para tratarmos da maneira certa as exceções que ocorrerem;
- Estudamos as principais palavras-chave quando o assunto é tratamento de erros em PHP 5: try, catch e throw;
- Estudamos superficialmente a classe Exception;
- Aprendemos a disparar o "sinal de alerta" aproveitando a condicional "if" e usando-a em conjunto com throw.

Aprendemos o funcionamento básico do "sistema" de manipulação dos erros no PHP 5:

Quando um erro (exceção) ocorre, este é "percebido" (com por exemplo a condicional "if"), instancia-se então a classe Exception (com "throw new Exception('msg', 01)") ou uma derivada, o controle é passado para o bloco "catch" correspondente, que por vez pode usar as informações que recebeu a respeito do erro na sua manipulação, como melhor convir.

Depois que todo o código do bloco catch que manipulou o erro é executado, o controle do código volta ao escopo mais geral (exceto se uma função exit() ou similar for executada dentro do bloco catch, o que terminaria a execução do código neste instante).

Encerrando

Espero sinceramente ter sido claro, e mesmo sem aprofundar muito neste estudo, ter passado o máximo de conhecimento a respeito dos fundamentos do tratamento de erro com exceções em PHP.

Continuemos a estudar e talvez um dia melhorar nosso país com o conhecimento!

Um abraço para todos da comunidade VOL!

L. Luiz

---

http://www.vivaolinux.com.br/artigo/Introducao-a-manipulacao-de-erros-em-PHP

9.11 - Trabalhando com Validação de Dados no PHP

Ótimo artigo em 5 partes na Revista do PHP de autoria do Er Abbott

- Validação de Formulários - 5º e última parte: Validando no servidor

- Validação de Formulários - Parte 04 (A fonteira cliente/servidor)

- Validação de Formulários - Parte 3: O baile de máscaras

- Validação de Formulários - Parte 2: Os Campos Especiais

- Validação de formulários - Parte 1 (O Planejamento)

Turbinando a Validação de Formulários

Validação de e-mails

Check valid e-mail

```
function esEmailValido($email)
{
    if (ereg("^[_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)*@([_a-zA-Z0-9-]+\.)*[a-zA-Z0-9-]
{2,200}\.[a-zA-Z]{2,6}$", $email ) )
        {
       return true;
    }
        else
        {
       return false;
    }
}
```

http://snippets.dzone.com/posts/show/4346

Tipos de Variáveis

```
<?php

function ss_array_as_string (&$array, $column = 0) {
    $str = "Array(<BR>\n";
    while(list($var, $val) = each($array)){
        for ($i = 0; $i < $column+1; $i++){
            $str .= "    ";
        }
        $str .= $var.' ==> ';
```

```
        $str .= ss_as_string($val, $column+1)."<BR>\n";
    }
    for ($i = 0; $i < $column; $i++){
        $str .= "    ";
    }
    return $str.')';
}

function ss_object_as_string (&$object, $column = 0) {
    if (empty($object->classname)) {
        return "$object";
    }
    else {
        $str = $object->classname."(<BR>\n";
        while (list(,$var) = each($object->persistent_slots)) {
            for ($i = 0; $i < $column; $i++){
                $str .= "    ";
            }
            global $$var;
            $str .= $var.' ==> ';
            $str .= ss_as_string($$var, column+1)."<BR>\n";
        }
        for ($i = 0; $i < $column; $i++){
            $str .= "    ";
        }
        return $str.')';
    }
}

function ss_as_string (&$thing, $column = 0) {
    if (is_object($thing)) {
        return ss_object_as_string($thing, $column);
    }
    elseif (is_array($thing)) {
        return ss_array_as_string($thing, $column);
    }
    elseif (is_double($thing)) {
        return "Double(".$thing.")";
    }
    elseif (is_long($thing)) {
        return "Long(".$thing.")";
    }
    elseif (is_string($thing)) {
        return "String(".$thing.")";
    }
    else {
        return "Unknown(".$thing.")";
    }
}

$my_variable=3;
//echo ss_as_string($my_variable);
echo ss_as_string($GLOBALS);
?>
```

Validação Usando Filtros do PHP

`Suportados pelo PHP com versão 5.2 ou superior.`

Constantes pré-definidas

As contantes abaixo são definidas por esta extensão e somente estarão disponíveis quando a extensão foi compilada com o PHP ou carregada dinamicamente durante a execução.

INPUT_POST (integer)
Variáveis POST.
INPUT_GET (integer)
Variáveis GET.
INPUT_COOKIE (integer)
Variáveis COOKIE.
INPUT_ENV (integer)
Variáveis ENV.
INPUT_SERVER (integer)
Variáveis SERVER.
INPUT_SESSION (integer)
Variáveis SESSION. (ainda não implementada)
INPUT_REQUEST (integer)
Variáveis REQUEST. (ainda não implementada)

FILTER_FLAG_NONE (integer)
Sem flags.
FILTER_REQUIRE_SCALAR (integer)
Flag usada para requerir escalar como entrada
FILTER_REQUIRE_ARRAY (integer)
Requer um array como entrada.
FILTER_FORCE_ARRAY (integer)
Sempre retorna um array.
FILTER_NULL_ON_FAILURE (integer)
Usa NULL ao invés de FALSE em falha.
FILTER_VALIDATE_INT (integer)
ID do filtro "int".
FILTER_VALIDATE_BOOLEAN (integer)
ID do filtro "boolean".
FILTER_VALIDATE_FLOAT (integer)
ID do filtro "float".
FILTER_VALIDATE_REGEXP (integer)
ID do filtro "validate_regexp".
FILTER_VALIDATE_URL (integer)
ID do filtro "validate_url".
FILTER_VALIDATE_EMAIL (integer)
ID do filtro "validate_email".
FILTER_VALIDATE_IP (integer)
ID do filtro "validate_ip".

FILTER_DEFAULT (integer)
ID do filtro padrão ("string").
FILTER_UNSAFE_RAW (integer)
ID do filtro "unsafe_raw".
FILTER_SANITIZE_STRING (integer)
ID do filtro "string".
FILTER_SANITIZE_STRIPPED (integer)
ID do filtro "stripped".
FILTER_SANITIZE_ENCODED (integer)
ID do filtro "encoded".
FILTER_SANITIZE_SPECIAL_CHARS (integer)
ID do filtro "special_chars".
FILTER_SANITIZE_EMAIL (integer)
ID do filtro "email".
FILTER_SANITIZE_URL (integer)
ID do filtro "url".
FILTER_SANITIZE_NUMBER_INT (integer)
ID do filtro "number_int".
FILTER_SANITIZE_NUMBER_FLOAT (integer)
ID of "number_float" filter.
FILTER_SANITIZE_MAGIC_QUOTES (integer)
ID do filtro "magic_quotes".
FILTER_CALLBACK (integer)
ID do filtro "callback".
FILTER_FLAG_ALLOW_OCTAL (integer)
Permite notação octal (*0[0-7]+*) no filtro "int".
FILTER_FLAG_ALLOW_HEX (integer)
Permite notação hexadecimal (*0x[0-9a-fA-F]+*) no filtro "int".
FILTER_FLAG_STRIP_LOW (integer)
Remove caracteres com valor ASCII menor que 32.
FILTER_FLAG_STRIP_HIGH (integer)
Remove caracteres com valor ASCII maior que 127.
FILTER_FLAG_ENCODE_LOW (integer)
Codifica caracteres com valor ASCII menor que 32.
FILTER_FLAG_ENCODE_HIGH (integer)
Codifica caracteres com valor ASCII maior que 127.
FILTER_FLAG_ENCODE_AMP (integer)
Codifica *&*.
FILTER_FLAG_NO_ENCODE_QUOTES (integer)
Não codifica ′ e ″.
FILTER_FLAG_EMPTY_STRING_NULL (integer)
(Nenhum uso no momento.)
FILTER_FLAG_ALLOW_FRACTION (integer)
Permite parte fracional no filtro "number_float".
FILTER_FLAG_ALLOW_THOUSAND (integer)
Permite separador de milhar (*,*) no filtro "number_float".
FILTER_FLAG_ALLOW_SCIENTIFIC (integer)
Permite notação científica (*e*, *E*) no filtro "number_float".

FILTER_FLAG_SCHEME_REQUIRED (integer)
Requer scheme no filtro "validate_url".
FILTER_FLAG_HOST_REQUIRED (integer)
Requer host no filtro "validate_url".
FILTER_FLAG_PATH_REQUIRED (integer)
Requer path no filtro "validate_url".
FILTER_FLAG_QUERY_REQUIRED (integer)
Requer query no filtro "validate_url".
FILTER_FLAG_IPV4 (integer)
Permite somente endereço IPv4 no filtro "validate_ip".
FILTER_FLAG_IPV6 (integer)
Permite somente endereço IPv6 no filtro "validate_ip".
FILTER_FLAG_NO_RES_RANGE (integer)
Não permite endereços reservados no filtro "validate_ip".
FILTER_FLAG_NO_PRIV_RANGE (integer)
Não permite endereços privados no filtro "validate_ip".

Funções de Filtragem

filter_has_var — Verifica se a variável é de um especificado tipo existente

- filter_id — Retorna o ID de um dado nome de filtro
- filter_input_array — Obtem variáveis externas e opcionalmente as filtra
- filter_input — Obtem a específica variável externa pelo nome e opcionalmente a filtra
- filter_list — Retorna a lista de todos filtros suportados
- filter_var_array — Obtêm múltiplas variáveis e opcionalmente as filtra
- filter_var — Filtra a variável com um especificado filtro

Detectando o tipo de variável:

```
$_GET['test'] = 1;
echo filter_has_var(INPUT_GET, 'test') ? 'Sim' : 'Não';

Retornando uma lista com os tipos de filtros suportados
```

```
<?php
print_r(filter_list());
?>
```

Filter_input:

```
$search_html = filter_input(INPUT_GET, 'search', FILTER_SANITIZE_SPECIAL_CHARS);
$search_url = filter_input(INPUT_GET, 'search', FILTER_SANITIZE_ENCODED);
echo "You have searched for $search_html.\n";
echo "<a href='?search=$search_url'>Search again.</a>";
```

filter_var

var_dump(filter_var('bob@example.com', FILTER_VALIDATE_EMAIL));
var_dump(filter_var('example.com', FILTER_VALIDATE_URL, FILTER_FLAG_SCHEME_REQ
UIRED));

```
<?php
error_reporting(E_ALL | E_STRICT);
$data = array(
    'product_id'    => 'libgd<script>',
    'component'     => '10',
    'versions'      => '2.0.33',
    'testscalar'    => array('2', '23', '10', '12'),
    'testarray'     => '2',
);

$args = array(
    'product_id'   => FILTER_SANITIZE_ENCODED,
    'component'    => array('filter'    => FILTER_VALIDATE_INT,
                            'flags'     => FILTER_FORCE_ARRAY,
                            'options'   => array('min_range' => 1,
'max_range' => 10)
                           ),
    'versions'     => FILTER_SANITIZE_ENCODED,
    'doesnotexist' => FILTER_VALIDATE_INT,
    'testscalar'   => array(
                            'filter' => FILTER_VALIDATE_INT,
                            'flags'  => FILTER_REQUIRE_SCALAR,
                           ),
    'testarray'    => array(
                            'filter' => FILTER_VALIDATE_INT,
                            'flags'  => FILTER_FORCE_ARRAY,
                           )

);

$myinputs = filter_var_array($data, $args);

var_dump($myinputs);
echo "\n";
?>
```

```
Just a little filter to validate IP v4 & v6
This little script display result in function of the query
<?php
$ipv6="2a01:e35:aaa4:6860:a5e7:5ba9:965e:cc93";
$ipv4="82.237.3.3";
$fake = "3342423423";
$ipv4priv = "255.255.255.255";
```

```
$ipv6priv = "::1";
echo "<pre>";
echo $ipv4;
echo "<br />";
var_dump(filter_var($ipv4,FILTER_VALIDATE_IP));
echo "<br />";
echo $ipv6;
echo "<br />";
var_dump(filter_var($ipv6,FILTER_VALIDATE_IP));
echo "<br />";
echo $fake;
echo "<br />";
var_dump(filter_var($fake,FILTER_VALIDATE_IP));
echo "<br />";
echo $ipv4priv;
echo "<br/>";
echo "FILTER_VALIDATE_IP, FILTER_FLAG_NO_RES_RANGE";
echo "<br />";
var_dump(filter_var($ipv4priv,FILTER_VALIDATE_IP,
FILTER_FLAG_NO_RES_RANGE));
echo "<br />";
echo $ipv4priv;
echo "<br/>";
echo "FILTER_FLAG_NO_PRIV_RANGE";
echo "<br />";
var_dump(filter_var($ipv4priv,FILTER_FLAG_NO_PRIV_RANGE));
echo "<br />";
echo $ipv6priv;
echo "<br/>";
echo "FILTER_FLAG_NO_PRIV_RANGE";
echo "<br />";
var_dump(filter_var($ipv6priv,FILTER_FLAG_NO_PRIV_RANGE));
echo "<br />";
echo $ipv6priv;
echo "<br />";
var_dump(filter_var($ipv6priv,FILTER_VALIDATE_IP));

echo "</pre>";
```

9.12 – Trabalhando com XML em PHP

XML no Manual do PHP - http://www.php.net/manual/pt_BR/book.xml.php
Introdução ao XML - http://www.criarweb.com/manuais/24

Lendo extrutura completa de arquivo XML:

data.xml

```
<?xml version='1.0'?>
<!DOCTYPE chapter SYSTEM "/just/a/test.dtd" [
<!ENTITY plainEntity "FOO entity">
<!ENTITY systemEntity SYSTEM "xmltest2.xml">
]>
<chapter>
 <TITLE>Titulo</TITLE>
 <para>
  <informaltable>
   <tgroup cols="3">
    <tbody>
     <row><entry>a1</entry><entry morerows="1">b1</entry><entry>c1</entry></row>
     <row><entry>a2</entry><entry>c2</entry></row>
     <row><entry>a3</entry><entry>b3</entry><entry>c3</entry></row>
    </tbody>
   </tgroup>
  </informaltable>
 </para>
 &systemEntity;
 <section id="about">
  <title>Sobre este documento</title>
  <para>
   <!-- this is a comment -->
   <?php echo 'Hi!  This is PHP version '.phpversion(); ?>
  </para>
 </section>
</chapter>
```

ler_xml.php

```
<?php
//Exemplo de Entity externa

$file = "data.xml";

function trustedFile($file) {
   // only trust local files owned by ourselves
   if (!eregi("^([a-z]+)://", $file)
       && fileowner($file) == getmyuid()) {
           return true;
   }
```

```
  return false;
}

function startElement($parser, $name, $attribs) {
  echo "&lt;<font color=\"#0000cc\">$name</font>";
  if (sizeof($attribs)) {
     while (list($k, $v) = each($attribs)) {
        echo " <font color=\"#009900\">$k</font>=\"<font
              color=\"#990000\">$v</font>\"";
     }
  }
  echo "&gt;";
}

function endElement($parser, $name) {
  echo "&lt;/<font color=\"#0000cc\">$name</font>&gt;";
}

function characterData($parser, $data) {
  echo "<b>$data</b>";
}

function PIHandler($parser, $target, $data) {
  switch (strtolower($target)) {
     case "php":
        global $parser_file;
        // If the parsed document is "trusted", we say it is safe
        // to execute PHP code inside it.  If not, display the code
        // instead.
        if (trustedFile($parser_file[$parser])) {
           eval($data);
        } else {
           printf("Untrusted PHP code: <i>%s</i>",
                 htmlspecialchars($data));
        }
        break;
  }
}

function defaultHandler($parser, $data) {
  if (substr($data, 0, 1) == "&" && substr($data, -1, 1) == ";") {
     printf('<font color="#aa00aa">%s</font>',
           htmlspecialchars($data));
  } else {
     printf('<font size="-1">%s</font>',
           htmlspecialchars($data));
  }
}
```

```
function externalEntityRefHandler($parser, $openEntityNames, $base, $systemId,
                                  $publicId) {
  if ($systemId) {
    if (!list($parser, $fp) = new_xml_parser($systemId)) {
      printf("Could not open entity %s at %s\n", $openEntityNames,
             $systemId);
      return false;
    }
    while ($data = fread($fp, 4096)) {
      if (!xml_parse($parser, $data, feof($fp))) {
        printf("XML error: %s at line %d while parsing entity %s\n",
               xml_error_string(xml_get_error_code($parser)),
               xml_get_current_line_number($parser), $openEntityNames);
        xml_parser_free($parser);
        return false;
      }
    }
    xml_parser_free($parser);
    return true;
  }
  return false;
}

function new_xml_parser($file) {
  global $parser_file;

  $xml_parser = xml_parser_create();
  xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, 1);
  xml_set_element_handler($xml_parser, "startElement", "endElement");
  xml_set_character_data_handler($xml_parser, "characterData");
  xml_set_processing_instruction_handler($xml_parser, "PIHandler");
  xml_set_default_handler($xml_parser, "defaultHandler");
  xml_set_external_entity_ref_handler($xml_parser, "externalEntityRefHandler");

  if (!($fp = @fopen($file, "r"))) {
    return false;
  }
  if (!is_array($parser_file)) {
    settype($parser_file, "array");
  }
  $parser_file[$xml_parser] = $file;
  return array($xml_parser, $fp);
}

if (!(list($xml_parser, $fp) = new_xml_parser($file))) {
  die("could not open XML input");
}
```

```
echo "<pre>";
while ($data = fread($fp, 4096)) {
   if (!xml_parse($xml_parser, $data, feof($fp))) {
      die(sprintf("XML error: %s at line %d\n",
                  xml_error_string(xml_get_error_code($xml_parser)),
                  xml_get_current_line_number($xml_parser)));
   }
}
echo "</pre>";
echo "parse complete\n";
xml_parser_free($xml_parser);
?>
```

Ler XML usando simplexml:

xml_lido.php

```
<?php
$xmlstr = <<<XML
<?xml version='1.0' encoding='ISO-8859-1' ?>
<filmes>
 <filme>
  <titulo>PHP: Iniciando o Parser</titulo>
  <personagens>
   <personagem>
   <nome>João de Brito</nome>
   <actor>Brito</actor>
   </personagem>
   <personagem>
   <nome>Manoel Cunha</nome>
   <actor>Manoel</actor>
   </personagem>
  </personagens>
  <comentario>
   O XML é uma linguagem. Ela é como uma linguagem de programação. Ou uma
   linguagem de script? Tudo será revelado após ler bem toda a
   documentação.
  </comentario>
  <votos type="thumbs">7</votos>
  <votos type="stars">5</votos>
 </filme>
</filmes>
XML;
?>
```

xml_ler.php

```
<?php
include 'xml_lido.php';
$xml = simplexml_load_string($xmlstr);
```

```
echo $xml->filme[0]->comentario; // "So this language. It's like..."
print '<br>';
echo $xml->filme[0]->titulo;
print '<br>';
echo $xml->filme[0]->personagens[0]->personagem[0]->nome;
print '<br>';
echo $xml->filme[0]->votos[0];
print '<br>';
echo $xml->filme[0]->votos[1];
?>
```

9.13 – Trabalhando com Constantes Mágicas e Superglonais em PHP

Variáveis do servidor

$_SERVER

Este é um array (vetor) 'superglobal', ou automaticamente global. Isto significa que ele é disponível em todos os escopos (níveis) de um script. Você não precisa fazer um: ... global $_SERVER; ... para poder acessá-lo dentro de funções ou métodos, como era necessário com $HTTP_SERVER_VARS. O array superglobal $_SERVER existe em qualquer sessão PHP e já contém um conjunto de chaves (índices) pré definidos e valorados. Os índices mais importantes são:

'REQUEST_URI'

O URI fornecido para acessar a página atual, por exemplo, '/index.html'.

'SCRIPT_NAME'

Contém o caminho completo do script atual. Útil para páginas que precisam apontar para elas mesmas (dinamicamente). A constante __FILE__ contém o caminho completo e nome do arquivo (mesmo incluído) atual.

'PHP_SELF'

O nome do arquivo do script atualmente em uso, relativo ao document root. Por exemplo, $_SERVER['PHP_SELF'] em um script com o endereço http://example.com/test.php/foo.bar pode ser /test.php/foo.bar. A constante __FILE__ contém o caminho completo e nome do arquivo (mesmo incluído) atual.

Se estiver rodando o PHP em linha de comando, esta variável não está disponível.

'SERVER_NAME'

O nome host do servidor onde o script atual é executado. Se o script está rodando em um host virtual, este será o valor definido para aquele host virtual.

'REQUEST_METHOD'

Contém o método de request utilizando para acessar a página. Geralmente 'GET', 'HEAD', 'POST' ou 'PUT'.

'QUERY_STRING'

A query string (string de solicitação), se houver, pela qual a página foi acessada.

'DOCUMENT_ROOT'

O diretório raiz sob onde o script atual é executado, como definido no arquivos de configuração do servidor.

'SCRIPT_FILENAME'

O caminho absoluto o script atualmente em execução.

Nota: Se o script for executado pela CLI com um caminho relativo, como file.php ou ../file.php, $_SERVER['SCRIPT_FILENAME'] irá conter o caminho relativo especificado pelo usuário.

Exemplos

```
$current_script = dirname($_SERVER['SCRIPT_NAME']);

$current_path  = dirname($_SERVER['SCRIPT_FILENAME']);

$request_uri = $_SERVER['REQUEST_URI'];

// Pick the predefined variable that works on your server
     return $_ENV['SCRIPT_URL'];

$_SERVER['QUERY_STRING'])

      $sPathPS = $_SERVER[PHP_SELF];
       $sPathFS = __FILE__;

echo 'http';
if($_SERVER['HTTPS']=='on'){echo 's';}
echo '://'.$_SERVER['SERVER_PORT'].$_SERVER['SCRIPT_NAME'];
if($_SERVER['QUERY_STRING']>' '){echo '?'.$_SERVER['QUERY_STRING'];}
```

Constantes Mágicas

```
__LINE__         A linha atual do script.

__FILE__        O caminho completo e nome do arquivo. Se utilizado dentro de um
include, o nome do arquivo incluído será retornado.

__FUNCTION__    O nome da função (Acrescentado no PHP 4.3.0). A partir do PHP 5
esta constante retorna o nome da função como ela foi
   declarada (sensível a maiúsculas e minúsculas). No PHP 4 sempre retorna o
nome em minúsculas.

__CLASS__       O nome da classe (Acrescentado no PHP 4.3.0). A partir do PHP 5
esta constante retorna o nome da função como ela foi
   declarada (sensível a maiúsculas e minúsculas). No PHP 4 sempre retorna o
nome em minúsculas.

__METHOD__      O nome do método de classe. (Acrescentado no PHP 5.0.0). O nome
do método é retornado como foi declarado (sensível a
   maiúsculas e minúsculas).

Exemplo:

if (realpath(__FILE__) == realpath($_SERVER['SCRIPT_FILENAME'])) {
  exit;
}
```

9.14 – Trabalhando com Formatação de Saída em PHP

Temos as três funções - printf, sprintf e vprintf

printf -- Mostra uma string formatada
void printf (string format [, mixed args])

sscanf -- Interpreta a entrada de uma string de acordo com um formato
mixed sscanf (string str, string formato [, string var1])

```
<?php
// Pegando o número serial
$serial = sscanf("SN/2350001","SN/%d");
// e a data de criação
$mandate = "January 01 2000";
list($month, $day, $year) = sscanf($mandate,"%s %d %d");
echo "O Item $serial foi criado em: $year-".substr($month,0,3)."-$day\n";
?>
```

Se parâetros opcionais são passados, a função retornará o número de valores assumidos. Os parâmetros opcionais devem ser passados por referência.

Exemplo 2. sscanf() - usando parâmetros opcionais

```
<?php
// pega informação do autor e gera uma entrada de DocBook
$auth = "24\tLewis Carroll";
$n = sscanf($auth,"%d\t%s %s", &$id, &$first, &$last);
echo "<author id='$id'>
  <firstname>$first</firstname>
  <surname>$last</surname>
</author>\n";
?>
```

fscanf -- Interpreta a leitura de um arquivo de acordo com um formato
mixed fscanf (resource handle, string formato [, string var1])

```
$handle = fopen ("users.txt","r");
while ($userinfo = fscanf ($handle, "%s\t%s\t%s\n")) {
  list ($name, $profission, $countrycode) = $userinfo;
  //... fazer algo com os valores
}
fclose($handle);

$goodevil = array ('There is a difference between %s and %s', 'good', 'evil');
echo call_user_func_array('sprintf',$goodevil);

<?php
$heading1 = "Label 1";
```

```
$heading2 = "Label 2";

$value1 = "31298";
$value2 = "98";

print "<pre>\n";
printf ("%'.-15.15s%'.6.6s\n", $heading1, $value1);
printf ("%'.-15.15s%'.6.6s\n", $heading2, $value2);
print "</pre>\n";
?>

<?php $f='<?php $f=%c%s%c; printf($f,39,$f,39); ?>'; printf($f,39,$f,39); ?>
```

sprintf -- Retorna uma string formatada
string sprintf (string format [, mixed args])

Um especificador de tipo que diz que o argumento deve ser tratado como do tipo. Os tipos possivéis são:

% - Um caractere porcento. Não é requerido neenhum argumento.
b - O argumento é tratado com um inteiro, e mostrado como um binário.
c - O argumento é tratado como um inteiro, e mostrado como o caractere ASCII correspondente.
d - O argumento é tratado como um inteiro, e mostrado como um número decimal com sinal.
u - O argumento é tratado com um inteiro, e mostrado como um número decimal sem sinal.
f - O argumento é tratado como um float, e mostrado como um número de ponto flutuante.
o - O argumento é tratado com um inteiro, e mostrado como un número octal.
s - O argumento é tratado e mostrado como uma string.
x - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras minúsculas).
X - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras maiúsculas).

```
<?php
$format = "There are %d monkeys in the %s";
printf($format,$num,$location);
?>
```

Este deve mostrar, "There are 5 monkeys in the tree". Mas imagine que nós estejamos criando a string de formatação em um arquivo separado, normalmente para internacionalizar e rescrevemos como:

Exemplo 2. Troca de argumentos

```
<?php
$format = "The %s contains %d monkeys";
printf($format,$num,$location);
?>
```

Agora nós temos um problema. A ordem dos argumentos na string de formatação não combina com os argumentos no código. Nós gostariamos de deixar o código como esta e simplesmente indicar na

string de formatação quais argumentos pertencem aonde. Podemos escrever a string de formatação assim:

Exemplo 3. Troca de argumento
```
<?php
$format = "The %2\$s contains %1\$d monkeys";
printf($format,$num,$location);
?>
```
Um beneficio adicional disto é ue você pode repetir os especificadores de conversão sem adicionar mais argumentos em seu código. Por exemplo:

Exemplo 4. Troca de argumento
```
<?php
$format = "The %2\$s contains %1\$d monkeys.
        That's a nice %2\$s full of %1\$d monkeys.";
printf($format, $num, $location);
?>
```

Veja também printf(), sscanf(), fscanf(), vsprintf() e number_format().
Examplos
Exemplo 5. sprintf(): inteiros prenchidos com zero
```
<?php
$isodate = sprintf("%04d-%02d-%02d", $year, $month, $day);
?>
```
Exemplo 6. sprintf(): formatando dinheiro
```
<?php
$money1 = 68.75;
$money2 = 54.35;
$money = $money1 + $money2;
// echo $money irá mostrar "123.1";
$formatted = sprintf("%01.2f", $money);
// echo $formatted irá mostrar "123.10"
?>
```

vprintf -- Mostra uma string formatada
void vprintf (string formato, array args)

Mostra uma string formatada de acordo com o formato (o qual é descrito na documentação para a função sprintf()).

Funciona como printf() mas aceita uma matriz de argumentos, ao invés de um número variavel de argumentos.

```
<?php
$fruits = array(1, 'banana',1, 'apples', 3, 'oranges', 2, 'peaches');

vprintf("I have %d %s, %d %s, %d %s and %d %s.", $fruits);
?>
```

9.15 – Trabalhando com Imagens e Gráficos em PHP

- 1 Trabalhando com a biblioteca gráfica GD
- 2 Gerando Thumbnails com GD
- 3 Gerando Imagens Dinamicamente
- 4 Desenhando retângulos
- 5 Desenhando polígonos
- 6 Desenhando arcos
- 7 Gerando Gráficos em PHP com a Biblioteca JPGraph
    - 7.1 O que é a JpGrapf?
    - 7.2 Requisitos
    - 7.3 Parâmetros de Compilação
- 8 Referência

Trabalhando com a biblioteca gráfica GD

(Se no Windows remover o ponto-e-vírgula ";" da linha "extension=php_gd.dll" do php.ini)

Gerando Thumbnails com GD

Artigo do BOZO no PHPBrasil:

http://phpbrasil.com/articles/article.php/id/1350

Gerando Imagens Dinamicamente

por Luiz Ribeiro

O PHP oferece uma interface ao módulo GD de Thomas Boutell. Usando tal módulo, você pode criar e editar imagens nos formatos JPEG e PNG. O formato GIF já foi aceito, mas como o algoritmo de compressão do GIF (LZW) contém uma patente de posse da Unisys, os desenvolvedores do módulo foram obrigados à retirar o suporte a esse formato nas versões mais recentes.

Bom, para iniciar vou explicar o procedimento para criar uma imagem usando o módulo GD em PHP. Se você não tem esse módulo, você pode fazer o download dele em http://www.boutell.com/gd/. Normalmente a GD acompanha uma instalação completa do PHP.

Para se criar a imagem, será usada a função ImageCreate(), então serão realizadas as alterações na imagem, então será finalizada a imagem usando ImageJpeg(), ImagePng() ou até ImageGif() se a versão do módulo GD for inferior à 1.4.

Bom, vamos ao que interessa. Primeiramente vamos criar uma pequena imagem com o seguinte texto: PHPBrasil. O código ficará da seguinte forma:

```
<?php
header("Content-type: image/gif"); //Informa ao browser que o arquivo é uma
imagem no formato GIF

$imagem = ImageCreate(150,40); //Cria uma imagem com as dimensões 100x20

$vermelho = ImageColorAllocate($imagem, 255, 0, 0); //Cria o segundo plano da
imagem e o configura para vermelho
$branco = ImageColorAllocate($imagem, 255, 255, 255); //Cria a cor de primeiro
plano da imagem e configura-a para branco

ImageString($imagem, 3, 3, 3, "PHPBrasil", $branco);
//Imprime na imagem o texto PHPBrasil na cor branca que está na variável $branco

ImageGif($imagem); //Converte a imagem para um GIF e a envia para o browser

ImageDestroy($imagem); //Destrói a memória alocada para a construção da imagem
GIF.
?>
```

Bom, o script está todo comentado e acho que você entendeu. Se alguma dúvida ficar martelando aí, manda um comentário. =D

Bom, neste exemplo usamos a função ImageGif() para converter a imagem, $imagem, e depoisa enviamos ao navegador. Mas poderíamos ter salvo esta imagem em um arquivo, ao invés de mostrar ela no navegador. Veja o exemplo:

```
<?php
$arquivo = "imagem1.gif";

$imagem = ImageCreate(150,40);

$vermelho = ImageColorAllocate($imagem, 255, 0, 0);
$branco = ImageColorAllocate($imagem, 255, 255, 255);

ImageString($imagem, 3, 3, 3, "PHPBrasil", $branco);
ImageGif($imagem, $arquivo);

ImageDestroy($imagem);

echo "A imagem foi salva no arquivo $arquivo.";
?>
```

Como você deve ter notado, apenas retiramos aquele header() (que informava ao browser que o arquivo era uma imagem), afinal este exemplo não irá mostrar a imagem no navegador e sim gravar ela em $arquivo, e também mudamos os parâmetros da função ImageGif() para salvar a imagem no arquivo.

Nesta parte do artigo, irei explicar como desenhar retângulos, polígonos e arcos.

Desenhando retângulos

Vamos ao primeiro exemplo, que irá desenhar um simples retângulo preenchido usando GD (o formato da imagem a seguir é PNG).

```
<?php
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
ImageFilledRectangle($imagem, 5, 10, 60, 14, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Bom, neste exemplo só há uma função nova, a função ImageFilledRectangle() que como seu próprio nome diz é uma função que cria um retângulo com as dimensões e posição informadas, e na cor azul, que foi definida na variável $azul.

Já para criar um retângulo sem preenchimento você simplesmente irá trocar a função ImageFilledRectangle() por ImageRectangle(). O exemplo ficará da seguinte forma:

```
<?php
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
ImageRectangle($imagem, 5, 10, 60, 14, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Como foi dito, este exemplo irá criar uma imagem com um retângulo sem preenchimento, mas sua borda terá a cor $azul.

Desenhando polígonos

Para desenhar polígonos, vamos usar a função ImagePolygon(), que irá criar um polígono sem preenchimento, e a função ImageFilledPolygon() que irá desenhar um polígono com preenchimento.

Em nosso primeiro exemplo, vamos desenhar um polígono com vértices de (12, 10), (15, 20), (50, 17) e (70, 10) com uma borda de azul-claro:

```
<?php
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
$pontos = array(12, 10, 15, 20, 50, 17, 70, 10);
ImagePolygon($imagem, $pontos, 4, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Bom, agora vamos criar um polígono preenchido, você já deve ter pensado que o código será o mesmo, mas ao invés de ImagePolygon() usaremos ImageFilledPolygon(), se você énsou isso, acertou em cheio. Vamos ver como ficaria nossa imagem com um retângulo preenchido:

```
<?php
```

```
header("Content-type: image/png");
$imagem = ImageCreate(100, 20);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
$azul = ImageColorAllocate($imagem, 20, 93, 233);
$pontos = array(12, 10, 15, 20, 50, 17, 70, 10);
ImageFilledPolygon($imagem, $pontos, 4, $azul);
ImagePng($imagem);
ImageDestroy($imagem);
?>
```

Desenhando arcos

Bom, agora vamos desenhar alguns arcos em nossas imagens, para isso vamos usar a função ImageArc(). Antes de começarmos, vou passar a sintaxe da função:

int ImageArc(int im, int cx, int cy, int w, int h, int s, int e, int co1);

Esta função desenha um arco em uma imagem, im, com uma posição inicial de X de cx e uma posição inicial Y de cy. O arco é de largura w e altura h, com um ângulo inicial de s e um ângulo final de e, tudo na cor co1.

Agora que já entendemos a função ImageArc() vamos ao nosso primeiro exemplo que irá desenhar uma elipse:

```
<?php
header("Content-type: image/gif");
$imagem = ImageCreate(500, 100);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
ImageColorTransparent($imagem, $branco);
$vermelho = ImageColorAllocate($imagem, 20, 93, 233);
ImageArc($imagem, 40, 50, 50, 40, 0, 360, $vermelho);
ImageGif($imagem);
ImageDestroy($imagem);
?>
```

O código acima funciona, pois para ter uma elipse, você precisa de uma diferença de 360 graus entre a posição inicial e a posição final. Aplicando esse conhecimento, também podemos desenhar um círculo preenchido usando a função ImageFillToBorder(). (Note que isso é um círculo, não uma elipse, porque os parâmetros de largura e altura têm o mesmo valor.)

```
<?php
header("Content-type: image/gif");
$imagem = ImageCreate(500, 100);
$branco = ImageColorAllocate($imagem, 255, 255, 255);
ImageColorTransparent($imagem, $branco);
$vermelho = ImageColorAllocate($imagem, 20, 93, 233);
ImageArc($imagem, 40, 50, 50, 50, 0, 360, $vermelho);
ImageFillToBorder($imagem, 50, 40, $vermelho);
ImageGif($imagem);
ImageDestroy($imagem);
?>
```

Observação: os exemplos acima foram retirados do livro PHP: Guia do Desenvolvedor que está na

lista de livros recomendados da PHPBrasil. Nos exemplos só foram alterados os nomes de algumas variáveis para facilitar a compreensão.

Bom, esse é o básico do módulo GD. Você com certeza tem muito a explorar ainda, em breve vou trazer mais alguns artigos sobre o assunto, para os que se interessaram, ou não entenderam alguma função podem ver no manual do PHP todas as funções de imagem:
http://br.php.net/manual/pt_BR/ref.image.php

Gerando Gráficos em PHP com a Biblioteca JPGraph

O que é a JpGrapf?

A JpGraph é uma biblioteca orientada a objetos de criação de gráficos, inteiramente escrita em PHP, que tem como base a extensão GD2 ou GD1 que acompanha o PHP.

Pode ser utilizada para criar diversos tipos de gráficos, de maneira fácil e escrevendo um mínimo de código. Quando aliada a bancos de dados torna os gráficos ainda mais interessantes.

Requisitos

Apache (http://httpd.apache.org)

PHP (www.php.net)

JpGraph (http://www.aditus.nu/jpgraph/)

Parâmetros de Compilação

- Compilar o PHP com suporte a GD e às fontes TTF (Linux):

Sugestão da documentação oficial da JpGrapf:

```
./configure --prefix=/usr/share \
--datadir=/usr/share/php \
--with-apxs=/usr/sbin/apxs \
--libdir=/usr/share \
--includedir=/usr/include \
--bindir=/usr/bin \
--with-config-file-path=/etc \
--enable-mbstring --enable-mbregex \
--with-pdflib=/usr \
--with-mysql  \
--with-ttf-dir=/usr/lib \
--with-freetype-dir=/usr/lib \
--with-gd --enable-gd-imgstrttf --enable-gd-native-ttf \
--with-zlib-dir=/usr/lib \
--with-png-dir=/usr/lib --with-jpeg-dir=/usr/lib --with-xpm-dir=/usr/X11R6 \
--with-tiff-dir=/usr/lib \
--enable-ftp \
--enable-memory-limit --enable-safe-mode \
--bindir=/usr/bin \
--enable-bcmath -enable-calendar \
--enable-ctype --with-ftp \
```

```
--enable-magic-quotes \
--enable-inline-optimization \
--with-bz2 \
--with-iconv
```

- No Windows basta descomentar no php.ini o suporte à GD2.
Obs.: No código de um gráfico não pode haver nenhuma saída em HTML ou texto.

Captura do site oficial.

Versões

Para PHP4 é indicada a versão 1.19 da JpGrapf e para a versão 5 do PHP é indicada a versão 2.0 beta.

Download das Fontes TTF (Linux)

http://corefonts.sourceforge.net/

http://www.gnome.org/fonts/

Como saber se o PHP já tem o suporte à JpGrapf?

Executar a função phpinfo(), que deve retornar:

- GD support - enabled

- FreeTypeSupport – enabled

- JPG support – enabled

- PNG support – enabled

- WBMP support – enabled

Exemplos que podem ser encontrados no site official em

http://www.aditus.nu/jpgraph/pdf/jpgraphddda.pdf

a) Radar plot

b) 3D Pie plot

c) Text−logarithmic linear plot

d) Combining linear and scatter plot

e) Scientific style plot

f) Using a background image

Exemplos de Gráficos

Gráfico dos Poços perfurados pelo DNOCS de 1900 a 1999, por estado:

Este gráfico acessa um banco PostgreSQL. Num pequeno form alguém informa um ano entre 1900 e 1999 e recebe o gráfico correspondente. Caso não informe o ano retornará o gráfico de todos os anos.

Código abaixo:

```
//Pequneo form de consulta
<h2 align=center>Consulta de Poços - DNOCS</h2>
<form name=frmPocos action="barras_pocos.php" method="post">
<center>Que ano que deseja Consultar? (Todos = vazio)<input name="ano" size=10>
<input type=submit value=Consultar></center>
</form>

//Arquivo barras_pocos.php
<?php
// Inclusão da biblioteca
include ("jpgraph.php");
include ("jpgraph_bar.php");
// Conexão ao banco PostgreSQL e consulta
$db = pg_connect("host=10.0.0.100 dbname=banco port=5432 user=user
password=pass") or die(pg_last_error());
$ano=$_POST['ano'];
// Se não for informado o ano da pesquisa no form anterior, exibirá todos os
poços, caso contrário mostra
// somente os poços do ano solicitado
if ($ano == "")
$sql=pg_query($db,"SELECT estado, count(poco) as quant from
recursos_hidricos.pocos group by estado");
else
$sql=pg_query($db,"SELECT estado, count(poco) as quant from
recursos_hidricos.pocos where
recursos_hidricos.pocos.ano = $ano group by estado");
```

```
while($row = pg_fetch_array($sql)) {
$quant[] = $row[1];     //Este array ($quant[]) sera usado em um dos eixos
$estado[] = $row[0];   // Este em outro eixo
}
// Construção da base do gráfico
$graph = new Graph(650,350,"auto");
$graph->SetScale("textint"); //Exibir as escalas
$graph->img->SetMargin(50,50,70,50);  //Margens dos 4 lados
$graph->title->Set('DNOCS - Poços X Estado do ano de '.$ano); // Título do
gráfico
$graph->title->SetFont(FF_VERDANA, FS_BOLD, 16);//Fonte do título
$graph->AdjBackgroundImage(0.4,0.7,-1); //Tipo de background
$graph->xaxis->title->Set('Estados'); //Título do eixo X
$graph->xaxis->SetLabelAngle(30); //Ângulo dos labels do eixo X
$graph->xaxis->SetTickLabels('Estados');
$graph->xaxis->SetFont(FF_VERDANA, FS_BOLD);//Fonte para o título do eixo X
$graph->xaxis->SetTickLabels($estado);  // Recebe o array dos estados do banco
$graph->yaxis->title->Set('Poços');
$graph->yaxis->SetFont(FF_FONT1, FS_BOLD);
$graph->yaxis->title->SetFont(FF_FONT1, FS_BOLD);
$graph->SetShadow(); //Adicionar sombra ao gráfico

//Adicionar um tipo de gráfico (barras)
$bplot = new BarPlot($quant);  //Recebe o outro array do banco
$bplot->SetFillColor("lightgreen"); // Cor do gráfico
$bplot->value->Show();
$bplot->value->SetFont(FF_ARIAL,FS_BOLD); //Fonte
$bplot->value->SetAngle(45); //Ângulo
$bplot->value->SetColor("black","navy"); //Cores

$graph->Add($bplot); //Adicionar o gráfico à base
$graph->Stroke();
?>
```

Ao baixar a JpGraph e descompactar no diretório web, veja a documentação, que exibe inúmeros tipos de gráficos com seus respectivos códigos ao lado, como também o subdiretório samples que tem 337 exemplos de gráficos.

Referência

http://www.phpbrasil.com/articles/print.php/id/164

http://www.zend.com/zend/tut/tutsweat3.php (ótimo tutorial, em inglês)

http://phpbrasil.com/articles/print.php/id/315 (outro muito bom e em português)

http://www.phpfreaks.com/print.php?cmd=tutorial&tut_id=115 (este abordando uso do MySQL)

9.16 – Trabalhando com Números em PHP

Muito Cuidado ao Lidar com Números em Ponto Flutuante

Teste em PHP

```
<?php
echo (int) ((0.1 + 0.7 ) * 10);
?>
```

Agora teste isso:

echo (int) ((0.2 + 0.7) * 10);

Não conclua muito apressadamente que é deficiência do PHP.

Neste momento devemos ter conhecimento de como se comportam os números, especialmente os floats, que são normalizados pelo IEEE.

Teste em Java

```
class teste {
    public static void main(String[] args) {
        System.out.println((int) ((0.1 + 0.7 ) * 10)); //Display the string.
    }
}
```

Em Java também dá o mesmo resultado do PHP, o que leva a crer que a coisa não depende da linguagem mas das normas de como foram construídos os números pelo IEEE.

O Effective Java sugere que se use int, long ou BigDecimal para representar os valores monetários. A classe BigDecimal foi desenvolvida para resolver dois tipos de problemas associados a números de ponto flutuante (floats e doubles): primeiro, resolve o problema da inexatidão da representação de números decimais; segundo, pode ser usado para trabalhar com números com mais de 16 dígitos significativos. Em compensação, utilizar BigDecimal pode tornar o programa menos legível por não haver sobrecarga dos operadores matemáticos para ela, sendo necessário usar métodos da classe. Veja, por exemplo, como você faria o programa da listagem 1 com BigDecimal:

BigDecimal d1 = new BigDecimal("1.95");

BigDecimal d2 = new BigDecimal("1.03");

System.out.println(d1.subtract(d2));

Utilizar os primitivos normalmente é mais rápido e mais prático, mas o problema fica por conta da definição das casas decimais. Você pode controlar diretamente as casas decimais, por exemplo, utilizando como unidade para os valores o centavo ao invés de real. Um int ou um long passariam a representar a quantidade de centavos presentes no valor, e não a quatidade de reais. Por exemplo:

long l1 = 195;

long l2 = 103;

System.out.println(l1 ? l2);

Listagem 6: Programa da listagem 1 com long

As variáveis acima dizem que você tem 195 centavos (e não R$ 1,95) e vai gastar 103 centavos, e não R$ 1,03. No final você ficará com 92 centavos (e não R$ 0,92).

Agora veja as recomendações do manual do PHP

O tamanho de um float depende também da plataforma e é de 64bits no formato IEEE(*). Nunca compare números em ponto flutuante em igualdades, sob pena de cometer erros.

Teste com PostgreSQL

SELECT CAST((0.1 + 0.7)*10 AS INTEGER);

Este sim, retorna o valor esperado.

Em Java:

System.out.println(1.95 - 1.03); // Retorna errado e em PHP retorna OK.

Em Ruby

(1.8+0.1)==(1.9) retorna false

O mesmo ocorre em Phyton.

```
<?php
/*
   extenso.php
   Copyright (C) 2002 Lyma

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 2 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program; if not, write to the Free Software
   Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

   Lyma (lyma@lymas.com)
   http://lymas.com

   Esta função recebe um valor numérico e retorna uma string contendo o
   valor de entrada por extenso.
```

```
   entrada: $valor (use ponto para centavos.)
   Ex.:

   echo extenso("12428.12"); //retorna: doze mil, quatrocentos e vinte e oito reais e doze centavos

   ou use:
   echo extenso("12428.12", true); //esta linha retorna: Doze Mil, Quatrocentos E Vinte E Oito
Reais E Doze Centavos

   saída..: string com $valor por extenso em reais e pode ser com iniciais em maiúsculas (true) ou
não.

   ---------------------------------------------------------------------------------------------------
   Modificado por Claudio Monteoliva - claudio@simplesefacil.com
   também é possível passar o valor para a funcao com a vírgula decimal.
   exemplo: echo extenso("12428,12"); // o retorna é o mesmo que a passagem com ponto decimal
*/


function extenso($valor=0, $maiusculas=false)
{
   // verifica se tem virgula decimal
   if (strpos($valor,",") > 0)
   {
     // retira o ponto de milhar, se tiver
     $valor = str_replace(".","",$valor);

     // troca a virgula decimal por ponto decimal
     $valor = str_replace(",",".",$valor);
   }

   $singular = array("centavo", "real", "mil", "milhão", "bilhão", "trilhão", "quatrilhão");
   $plural = array("centavos", "reais", "mil", "milhões", "bilhões", "trilhões", "quatrilhões");

   $c = array("", "cem", "duzentos", "trezentos", "quatrocentos", "quinhentos", "seiscentos",
"setecentos", "oitocentos", "novecentos");
   $d = array("", "dez", "vinte", "trinta", "quarenta", "cinquenta", "sessenta", "setenta", "oitenta",
"noventa");
   $d10 = array("dez", "onze", "doze", "treze", "quatorze", "quinze","dezesseis", "dezesete",
"dezoito", "dezenove");
   $u = array("", "um", "dois", "três", "quatro", "cinco", "seis", "sete", "oito", "nove");

   $z=0;

   $valor = number_format($valor, 2, ".", ".");
   $inteiro = explode(".", $valor);
   for($i=0;$i<count($inteiro);$i++)
      for($ii=strlen($inteiro[$i]);$ii<3;$ii++)
```

```
        $inteiro[$i] = "0".$inteiro[$i];

    $fim = count($inteiro) - ($inteiro[count($inteiro)-1] > 0 ? 1 : 2);
    for ($i=0;$i<count($inteiro);$i++) {
        $valor = $inteiro[$i];
        $rc = (($valor > 100) && ($valor < 200)) ? "cento" : $c[$valor[0]];
        $rd = ($valor[1] < 2) ? "" : $d[$valor[1]];
        $ru = ($valor > 0) ? (($valor[1] == 1) ? $d10[$valor[2]] : $u[$valor[2]]) : "";

        $r = $rc.(($rc && ($rd || $ru)) ? " e " : "").$rd.(($rd &&
$ru) ? " e " : "").$ru;
        $t = count($inteiro)-1-$i;
        $r .= $r ? " ".($valor > 1 ? $plural[$t] : $singular[$t]) : "";
        if ($valor == "000")$z++; elseif ($z > 0) $z--;
        if (($t==1) && ($z>0) && ($inteiro[0] > 0)) $r .= (($z>1) ? " de " : "").$plural[$t];
        if ($r) $rt = $rt . ((($i > 0) && ($i <= $fim) &&
($inteiro[0] > 0) && ($z < 1)) ? ( ($i < $fim) ? ", " : " e ") : " ") . $r;
    }

        if(!$maiusculas){
                return($rt ? $rt : "zero");
        } else {
                return (ucwords($rt) ? ucwords($rt) : "Zero");
        }

}

//echo extenso("12428,12");

//Completar com zeros à esquerda de um numero até que fique com o numero de caracteres $X,
// retornando como Caráter

function ZeroEsquerda($kNumero, $X)
{
 $NumStrZero = trim((string)$kNumero);
 $QuantosZeros = $X - strlen($NumStrZero);
 for ( $i = 1; $i <= $QuantosZeros; $i++ )
  {
    $NumStrZero = '0'.$NumStrZero;
  }
 return $NumStrZero;
}


?>
```

9.17 – Trabalhando com Permissões de Arquivos e Diretórios

chmod - altera permissões de arquivos e diretórios

```
<?php
chmod ("/arquivo/diretorio", 755);  // decimal; provavelmente incorreto
chmod ("/arquivo/diretorio", "u+rwx,go+rx"); // string; incorreto
chmod ("/arquivo/diretorio", 0755);  // octal; representaÃ§Ã£o correta do modo
?>

function permissoes($arquivo,$perms,$acao){
        print "<form name=frm method=post action=acoes.php>";
        print "<input name=pm value=$perms>";
        print "<input type=hidden name=perms value=$perms>";
        print "<input type=hidden name=ar value=$arquivo>";
        print "<input type=hidden name=acao value=$acao>";
        print "<input name=ar value=$arquivo readonly style='background-
color:#FAEBD7'>";
        print "<input type=submit name=prm value=Alterar>";
        print "</form>";

        if (isset($_POST['prm'])){
                $ar=$_POST['ar'];
                $perms=octdec($_POST['pm']);
                $ch = chmod($ar, $perms);
                if(!$ch) {
                        die ("Erro ao alterar as permissÃ□Âµes!");
                }else{
                        print "<script>location='index.php'</script>";
                }
        }
}

<?php
// Escrita e leitura para o proprietario, nada ninguem mais
chmod ("/somedir/somefile", 0600);

// Escrita e leitura para o proprietario, leitura para todos os outros
chmod ("/somedir/somefile", 0644);

// Tudo para o proprietario, leitura e execucao para os outros
chmod ("/somedir/somefile", 0755);

// Tudo para o proprietario, leitura e execucao para o grupo do prop
chmod ("/somedir/somefile", 0750);
?>

Value    Permission Level
400    Owner Read
200    Owner Write
100    Owner Execute
40    Group Read
20    Group Write
10    Group Execute
4    Global Read
```

```
2    Global Write
1    Global Execute
```

```
<?php
function chmodnum($mode) {
   $mode2=$mode;
   $realmode = "";
   $legal =  array("","w","r","x","-");
   $attarray = preg_split("//",$mode);
   for($i=0;$i<count($attarray);$i++){
       if($key = array_search($attarray[$i],$legal)){
           $realmode .= $legal[$key];
       }
   }
   $mode = str_pad($realmode,9,'-');
   $trans = array('-'=>'0','r'=>'4','w'=>'2','x'=>'1');
   $mode = strtr($mode,$trans);
   $newmode = '';
   $newmode .= $mode[0]+$mode[1]+$mode[2];
   $newmode .= $mode[3]+$mode[4]+$mode[5];
   $newmode .= $mode[6]+$mode[7]+$mode[8];
   return $mode2.' = '.$newmode;
}

echo chmodnum('drwxr-xr-x');
?>
```

```
alguns exemplos:

drwxr-xr-x => 755
drwxr-xr-x => 755
dr-xr-xr-x => 555
drwxr-xr-x => 755
drwxr-xr-x => 755
drwxr-xr-x => 755
drwxr-xr-x => 755
drwxrwxrwt => 776
drwxr-xr-x => 755
drwxr-xr-x => 755
lrwxrwxrwx => 777
```

chown

Esta função não trabalha com arquivos remotos

```
<?php

$file_name= "test";
$path = "/var/www/html/test/" . $file_name ;

$user_name = "root";

chown($path, $user_name);
```

```
?>

<?php
function recurse_chown_chgrp($mypath, $uid, $gid)
{
   $d = opendir ($mypath) ;
   while(($file = readdir($d)) !== false) {
       if ($file != "." && $file != "..") {

           $typepath = $mypath . "/" . $file ;

           //print $typepath. " : " . filetype ($typepath). "<BR>" ;
           if (filetype ($typepath) == 'dir') {
               recurse_chown_chgrp ($typepath, $uid, $gid);
           }

           chown($typepath, $uid);
           chgrp($typepath, $gid);

       }
   }

 }

recurse_chown_chgrp ("uploads", "ribafs", "meugrupo") ;
?>

<?php
function recurse_chown_chgrp($path2dir, $uid, $gid){
   $dir = new dir($path2dir);
   while(($file = $dir->read()) !== false) {
       if(is_dir($dir->path.$file)) {
           recurse_chown_chgrp($dir->path.$file, $uid, $gid);
       } else {
           chown($file, $uid);
           chgrp($file, $gid);
       }
   }
   $dir->close();
}
?>
```

chgrp -- Modifica o grupo do arquivo

filegroup -- Lê o grupo do arquivo

fileperms -- Lê as permissões do arquivo

fileowner -- Lê o dono (owner) do arquivo

is_readable -- Diz se o arquivo/diretório é legivel (readable)

```
<?php
if (is_readable('my_link')) {
```

```
  header('Location: /my_link');
}
?>
```

is_writable -- Diz se pode-se escrever para o arquivo (writable)

```
<?php

$file = '/home/vincent/arquivo.sh';

if(is_executable($file)) {
   echo $file.' Ã© executÃ¡vel';
} else {
   echo $file.' nÃ£o Ã© executÃ¡vel';
}

?>
```

umask -- Modificar a umask atual

```
<?php
umask(0670);                     //- set umask
$handle = fopen('file', 'w');  //- 0006
mkdir("/path/dir");            //- 0107
?>

calculate the result:
<?php
$umask = 0670;
umask($umask);
//- if you are creating a new directory, $permission = 0777;
//- if you are creating a new file, $permission = 0666.
printf( "result: %04o", $permission & ( 0777 - $umask) );
?>
```

9.18 – Trabalhando com Strings em PHP

- 1 substr -- Retorna uma parte de uma string
- 2 substr_replace
- 3 Encontrar Posição de caractere em String
- 4 Contando Ocorrências de Substring em String
- 5 Trocando Ponto por Vírgula e vice-versa
- 6 Conversão de Strings
- 7 Trabalhando com os Caracteres de Strings
- 8 Validação de Caracteres
- 9 ctype_alnum - Checa por caracteres alfanuméricos
- 10 ctype_alpha - Checa por caracteres alfabéticos
- 11 ctype_digit - Checa por caracteres numéricos
- 12 ctype_lower - Checa por caracteres minúsculos
- 13 ctype_punct - Checa por Caracteres que não sejam espaço em branco nem alfanuméricos
- 14 ctype_space - Checa por espaços em branco
- 15 Validação de Tipos
- 16 Cases
- 17 Índices com Str_Pad
- 18 String para TimeStamp

Retorna uma parte de uma string

string substr (string string, int start [, int length])

Exemplo 1. Uso básico de substr()

```
<?php
$rest = substr("abcdef", 1);    // retorna "bcdef"
$rest = substr("abcdef", 1, 3); // retorna "bcd"
$rest = substr("abcdef", 0, 4); // retorna "abcd"
$rest = substr("abcdef", 0, 8); // retorna "abcdef"

// Outra opção é acessar atravéz de chaves
$string = 'abcdef';
echo $string{0};                // retorna a
echo $string{3};                // retorna d
?>
```

Se start for negativo, a string retornada irá começar no caractere start a partir do fim de string.

Exemplo 2. Usando um inicio negativo

```
<?php
$rest = substr("abcdef", -1);    // retorna "f"
$rest = substr("abcdef", -2);    // retorna "ef"
$rest = substr("abcdef", -3, 1); // retorna "d"
?>
```

Exemplo 3. Usando um length negativo

```
<?php
$rest = substr("abcdef", 0, -1);  // retorna "abcde"
$rest = substr("abcdef", 2, -1);  // retorna "cde"
$rest = substr("abcdef", 4, -4);  // retorna ""
$rest = substr("abcdef", -3, -1); // retorna "de"
?>

<h2>Sobrescrevendo Strings</h2>

str_replace

str_replace --  Substitui todas as ocorrências da string de procura com a string
de substituição

mixed str_replace ( mixed pesquisa, mixed substitui, mixed assunto [, int
&count] )

<pre>
<?php
// Fornece: <body text='black'>
$bodytag = str_replace("%body%", "black", "<body text='%body%'>");

// Fornece: Hll Wrld f PHP
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");

// Fornece: você comeria pizza, cerveja e sorvete todos os dias
$frase  = "você comeria frutas, vegetais, e fibra todos os dias.";
$saudavel = array("frutas", "vegetais", "fibra");
$saboroso  = array("pizza", "cerveja", "sorvete");

$novafrase = str_replace($saudavel, $saboroso, $frase);

// Uso do parâmetro count está disponível no PHP 5.0.0
$str = str_replace("ll", "", "good golly miss molly!", $count);
echo $count; // 2
?>
```

substr_replace

substr_replace -- Substitui o texto dentro de uma parte de uma string

string substr_replace (string string, string replacement, int start [, int length])

```
<?php
$var = 'ABCDEFGH:/MNRPQR/';
echo "Original: $var<hr>\n";

/* Estes dois exemplos substituem tudo de $var com 'bob'. */
echo substr_replace($var, 'bob', 0) . "<br>\n";
echo substr_replace($var, 'bob', 0, strlen($var)) . "<br>\n";

/* Insere 'bob' direto no começo de $var. */
```

```
echo substr_replace($var, 'bob', 0, 0) . "<br>\n";

/* Estes dois exemplos substituem 'MNRPQR' em $var com 'bob'. */
echo substr_replace($var, 'bob', 10, -1) . "<br>\n";
echo substr_replace($var, 'bob', -7, -1) . "<br>\n";

/* Deleta 'MNRPQR' de $var. */
echo substr_replace($var, '', 10, -1) . "<br>\n";
?>
```

Encontrar Posição de caractere em String

strpos

strpos -- Encontra a posição da primeira ocorrência de uma string

int strpos (string str, string procurar [, int offset])

Exemplos strpos()

```
<?php
//$str = 'abc';
$str = 'cba';
$procurar  = 'a';
$posicao = strpos($str, $procurar);

// Note o uso de ===.  Simples == não funcionaria como esperado
// por causa da posição de 'a' é 0 (primeiro) caractere.
if ($pos === false) {
   echo "A string '$procurar' não foi encontrada na string '$str'";
} else {
   echo "A string '$procurar' foi encontrada na string '$str'";
   echo " e está na posição $posicao";
}

?>

<?php

//$email = 'ribafs@gmail.com.br';
$email = 'ribafs@gmail.com';
$usuario = substr ($email, 0, strpos ($email, '@'));
// Lembrando: substr ( string string, int start [, int length] )
$dominio = substr ($email, strpos ($email, '@')+1);
echo "Usuário '$usuario' e Domínio '$dominio'"; // o comprimento default é até o
final
?>
```

Contando Ocorrências de Substring em String

substr_count -- Conta o número de ocorrências de uma substring

int substr_count (string str, string conte_me)

substr_count() retorna o número de vezes que a substring conte_me ocorre na string str.

```
<?php
  $str = "Olá mundo do PHP";

  if (substr_count($str, "do") == 0)
   echo "nenhum";

  // same as:

  if (strpos($str, "do") === false)
   echo "nenhum";
?>
```

Exemplo 1. Exemplo substr_count()

```
<?php
print substr_count("This is a test", "is"); // mostra 2
?>
```

Trocando Ponto por Vírgula e vice-versa

Se temos campos tipo moeda, devemos exibir com vírgula e gravar no banco com ponto.

Para isso uma boa saída é usar a dupla de funções implode e explode.

Antes de exibir na tela (em consultas):

$f_custo_produtivo=explode(".",$f_custo_produtivo);

$f_custo_produtivo=implode(",",$f_custo_produtivo);

Antes de gravar no banco (inclusão e atualização):

$f_custo_produtivo=explode(",",$f_custo_produtivo);

$f_custo_produtivo=implode(".",$f_custo_produtivo);

Conversão de Strings

```
$foo = 1 + "10.5";echo $foo."<br>";             // $foo é float (11.5)
$foo = 1 + "-1.3e3";echo $foo."<br>";              // $foo é float (-1299)
$foo = 1 + "bob-1.3e3";echo $foo."<br>";          // $foo é integer (1)
$foo = 1 + "bob3";echo $foo."<br>";                // $foo é integer (1)
$foo = 1 + "10 Small Pigs";echo $foo."<br>";      // $foo é integer (11)
$foo = 4 + "10.2 Little Piggies";echo $foo."<br>"; // $foo é float (14.2)
$foo = "10.0 pigs " + 1;echo $foo."<br>";          // $foo é float (11)
$foo = "10.0 pigs " + 1.0;echo $foo."<br>";        // $foo é float (11)
```

Trabalhando com os Caracteres de Strings

```
// Pega o primeiro caracter da string
$str = 'Isto é um teste.';
$first = $str{0};
echo $first."<br>";
// Pega o terceiro caracter da string
$third = $str{2};
echo $third."<br>";
// Pega o último caracter da string
$str = 'Isto ainda é um teste.';
$last = $str{strlen($str)-1};
echo $last."<br>";
// Modifica o ultimo caracter da string
$str = 'Olhe o mal';
echo $str{strlen($str)-1} = 'r';
```

Validação de Caracteres

```
ctype_alnum
ctype_alpha
ctype_cntrl
ctype_digit
ctype_graph
ctype_lower
ctype_print
ctype_punct
ctype_space
ctype_upper
ctype_xdigit
```

ctype_alnum - Checa por caracteres alfanuméricos

```
$strings = array('AbCd1zyZ9', 'foo!#$bar');

foreach ($strings as $testcase) {
   if (ctype_alnum($testcase)) {
       echo "The string $testcase consists of all letters or digits.\n";
   } else {
       echo "The string $testcase does not consist of all letters or digits.\n";
   }
}
```

ctype_alpha - Checa por caracteres alfabéticos

```
$strings = array('KjgWZC', 'arf12');
foreach ($strings as $testcase) {
   if (ctype_alpha($testcase)) {
       echo "The string $testcase consists of all letters.\n";
   } else {
       echo "The string $testcase does not consist of all letters.\n";
   }
```

```
}
```

ctype_digit - Checa por caracteres numéricos

```
$strings = array('1820.20', '10002', 'wsl!12');
foreach ($strings as $testcase) {
   if (ctype_digit($testcase)) {
       echo "The string $testcase consists of all digits.\n";
   } else {
       echo "The string $testcase does not consist of all digits.\n";
   }
}
// Alerta: Ao executar veja que somente é válido quando todos são dígitos
// Não é indicado para testar valores decimais, com ponto ou vírgula
```

ctype_lower - Checa por caracteres minúsculos

```
$strings = array('aac123', 'qiutoas', 'QASsdks');
foreach ($strings as $testcase) {
   if (ctype_lower($testcase)) {
       echo "The string $testcase consists of all lowercase letters.\n";
   } else {
       echo "The string $testcase does not consist of all lowercase letters.\n";
   }
}
```

ctype_punct - Checa por Caracteres que não sejam espaço em branco nem alfanuméricos

```
$strings = array('ABasdk!@!$#', '!@ # $', '*&$()');
foreach ($strings as $testcase) {
   if (ctype_punct($testcase)) {
       echo "The string $testcase consists of all punctuation.\n";
   } else {
       echo "The string $testcase does not consist of all punctuation.\n";
   }
}
```

ctype_space - Checa por espaços em branco

Validação de Tipos

```
intval
is_array
```

```
is_bool
is_callable
is_double
is_float
is_int
is_integer
is_long
is_null
is_numeric
is_object
is_real
is_resource
is_scalar
is_string
isset
print_r
serialize
settype
strval
unserialize
unset
```

Cases

```
strtoupper($str) - tudo maiúsculo
strtolower($str) - tudo minúsculo
ucfirst($str) - Converte para maiúscula o primeiro caractere de uma STRING
ucwords($STR) - Converte para maiúsculas o primeiro caractere de cada PALAVRA
```

Índices com Str_Pad

str_pad -- Preenche uma string para um certo tamanho com outra string

string str_pad (string input, int pad_length [, string pad_string [, int pad_type]])

Exemplo:

$players =

```
  array("DUNCAN, king of Scotland"=>"Larry",
        "MALCOLM, son of the king"=>"Curly",
        "MACBETH"=>"Moe",
        "MACDUFF"=>"Rafael");
```

echo "

```
";

// Print a heading
echo str_pad("Dramatis Personae", 50, " ", STR_PAD_BOTH) . "\n";

// Print an index line for each entry
foreach($players as $role=>$actor)
  echo str_pad($role, 30, ".")
```

```
      . str_pad($actor, 20, ".", STR_PAD_LEFT) . "\n";

echo "

";
```

Resultado:

```
                    Dramatis Personae
DUNCAN, king of Scotland.....................Larry
MALCOLM, son of the king.....................Curly
MACBETH.......................................Moe
MACDUFF.....................................Rafael
```

String para TimeStamp

```
// Absolute dates and times
$var = strtotime("25 December 2002");
$var = strtotime("14/5/1955");
$var = strtotime("Fr1, 7 Sep 2001 10:28:07 -1000");

// The current time: equivalent to time(  )
$var = strtotime("now");

// Relative times
echo strtotime("+1 day");
echo strtotime("-2 weeks");
echo strtotime("+2 hours 2 seconds");

//Care should be taken when using strtotime( ) with user-supplied dates. It's
better to limit the use of strtotime( ) to cases when
//the string to be parsed is under the control of the script, for example,
checking a minimum age using a relative date:
// date of birth: timestamp for 16 August, 1983
$dob = mktime(0, 0, 0, 16, 8, 1982);

// Now check that the individual is over 18
if ((float)$dob < (float)strtotime("-18 years"))
  echo "Legal to drive in the state of Victoria";
```

9.19 – Trabalhando com URLs no PHP

Passando Parâmetros pela URL

Primeiro (âncora)

<a href="arquivo.php?parametro1=valor1&parametro2=valor2&parametro3=valor3">Link</a>

```
arquivo.php
$par1=$_GET['parametro1'];
$par2=$_GET['parametro2'];
$par3=$_GET['parametro3'];
```

Segundo (action de form)

```
<form name=frm method=post action="arquivo2.php?
parametro1=valor1&parametro2=valor2">
...
```

arquivo2.php

```
$par1=$_POST['parametro1'];
$par2=$_POST['parametro1'];
```

Terceiro (URL)

http://localhost/teste.php?parametro1=valor1

teste.php

$par1=$_GET['parametro1'];

Quarto (location no javascript)

```
<?php
// Já vindo de outro script, chamado via POST
$a = $_POST['a'];

?>
<script>
if(confirm("Confirma?")){
        location="vai.php?a='<?=$a?>' ";
}else{
        location='volta.php';
}
</script>
```

Reconstruct URL string in PHP

```
// find out the domain:
$domain = $_SERVER['HTTP_HOST'];
// find out the path to the current file:
$path = $_SERVER['SCRIPT_NAME'];
// find out the QueryString:
$queryString = $_SERVER['QUERY_STRING'];
```

```
 // put it all together:
 $url = "http://" . $domain . $path . "?" . $queryString;
 echo "The current URL is: " . $url . "
";

 // An alternative way is to use REQUEST_URI instead of both
 // SCRIPT_NAME and QUERY_STRING, if you don't need them seperate:
 $url2 = "http://" . $domain . $_SERVER['REQUEST_URI'];
 echo "The alternative way: " . $url2;
```

Do site - http://snippets.dzone.com/posts/show/4054

9.20 – Trabalhando com Criptografia no PHP

Classe usando a função crypt do PHP para criptografar e descriptografar uma script.
Fonte: http://phpro.org

```
<?php

// make it of break it
error_reporting(E_ALL);

/**
* Class to provide 2 way encryption of data
*
* @author    Kevin Waterson
* @copyright    2009    PHPRO.ORG
*
*/
class proCrypt
{
    /**
    *
    * This is called when we wish to set a variable
    *
    * @access    public
    * @param    string    $name
    * @param    string    $value
    *
    */
    public function __set( $name, $value )
    {
        switch( $name)
        {
            case 'key':
            case 'ivs':
            case 'iv':
            $this->$name = $value;
            break;

            default:
            throw new Exception( "$name cannot be set" );
        }
    }

    /**
    *
    * Gettor - This is called when an non existant variable is cal
led
```

```
    *
    * @access    public
    * @param    string    $name
    *
    */
    public function __get( $name )
    {
        switch( $name )
        {
            case 'key':
            return 'keee';

            case 'ivs':
            return mcrypt_get_iv_size( MCRYPT_RIJNDAEL_128, MCRYPT
_MODE_ECB );

            case 'iv':
            return mcrypt_create_iv( $this->ivs );

            default:
            throw new Exception( "$name cannot be called" );
        }
    }

    /**
    *
    * Encrypt a string
    *
    * @access    public
    * @param    string    $text
    * @return    string    The encrypted string
    *
    */
    public function encrypt( $text )
    {
        // add end of text delimiter
        $data = mcrypt_encrypt( MCRYPT_RIJNDAEL_128, $this->key, $
text, MCRYPT_MODE_ECB, $this->iv );
        return base64_encode( $data );
    }

    /**
    *
    * Decrypt a string
    *
    * @access    public
    * @param    string    $text
    * @return    string    The decrypted string
```

```
    *
    */
    public function decrypt( $text )
    {
        $text = base64_decode( $text );
        return mcrypt_decrypt( MCRYPT_RIJNDAEL_128, $this->key, $t
ext, MCRYPT_MODE_ECB, $this->iv );
    }
} // end of class
```

Exemplo de Uso

```
<?php

// a new proCrypt instance
$crypt = new proCrypt;

// encrypt the string
$encoded = $crypt->encrypt( 'my message');
echo $encoded."\n";

// decrypt the string
echo $crypt->decrypt( $encoded ) . "\n";

?>
```

Encode e Decode

PHP/MySQL - Função ()encode e ()decode

Fala galera! Nesse artigo estarei dando continuação ao assunto criptografia em PHP. Como vocês puderam observar na semana passada, eu expliquei a função password(), que tem por finalidade fazer a criptografia de senhas.

Uma desvantagem do uso dessa função, é que não podemos descriptografar o dado depois. A seguir, vou apresentar um outro método de criptografia que é o encode (codifica o dado) e o decode (decodifica o dado).

Exemplo prático

1. Vamos criar uma tabela (mensagem.php):

CREATE TABLE tb_mensagem (
id int(3) NOT NULL auto_increment,
de varchar(80) NOT NULL DEFAULT " ,
para varchar(80) NOT NULL DEFAULT " ,
mensagem text NOT NULL DEFAULT " ,

```
PRIMARY KEY (id)
);
```

2. "Popular" a nossa base de dados:

```
INSERT INTO tb_mensagem(de,para,mensagem)
VALUES('Júlio César Martini','iMasters',
encode('Mensagem sigilosa: Artigo sobre criptografia','teste'))
```

Resultado:

Id | De | Para | Mensagem
1 | Júlio César Martini | iMasters |$P²Ý5¾î_¯¹“ÁøJ®nzOAHG²³¼8¿ ê6êEòYÉ%O,{˜

Depois de ter efetuado este comando SQL, o resultado que você tem na sua tabela (tb_mensagem) é o apresentado acima.

Como você pode observar, o campo mensagem é o que recebeu a função encode(). Veja o monte de caracteres esquisitos que apareceram no lugar da mensagem original.

Agora vocês vão me perguntar, como eu faço para ele mostrar a mensagem certa? Para isso, vamos fazer uso da função decode().

3. Arquivo que conecta com a nossa base de dados MySQL:

```
<?
/* Conecta com um banco de dados MySQL conforme parâmetros enviados (servidor = localhost)
Banco de Dados: $dbname
Porta: $usuario
Senha: $senha*/

$dbname="nomedobd";
$usuario="";
$password="";

//1º passo - Conecta ao servidor MySQL
if(!($id = mysql_connect("localhost",$usuario,$password))) {
echo "<p align="center"><big><strong>Não foi possível estabelecer
uma conexão com o gerenciador MySQL. Favor Contactar o Administrador.
</strong></big></p>";
exit;
}

//2º passo - Seleciona o Banco de Dados
if(!($con=mysql_select_db($dbname,$id))) {
echo " <p align="center"><big><strong>Não foi possível estabelecer
uma conexão com o gerenciador MySQL. Favor Contactar o Administrador.
</strong></big></p>";
```

```
exit;
}
?>
```

4. Vamos criar uma página index.php que vai exibir a mensagem na tela:

```
<?
include "conecta.php"; //Arquivo que conecta com a nossa base de dados MySQL

$sql = mysql_query("SELECT de,para,decode(mensagem,'teste') FROM tb_mensagem") or
die("ERRO no SQL : ".mysql_error());
$array = mysql_fetch_array($sql);

echo $array['de']; echo "<br>";
echo $array['para']; echo "<br><br>";
echo $array['mensagem'];

?>
```

Na linha do SELECT, fazemos uso da função decode(), que vai decodificar aquela mensagem para que ela possa ser exibida corretamente na tela para o usuário.

Como vocês podem observar nessa linha: decode(mensagem,'teste'), passamos a chave para decodificar a mensagem. Nesse caso é teste.

Lembre-se que na hora de fazer o encode(), definimos teste como sendo a chave. Então, para decodificar, precisamos informar ela novamente.

Convido o leitor a tirar o decode() do SQL e ver o resultado.

A linha vai ficar assim:

```
<?
...
$sql = mysql_query("SELECT de,para,mensagem) FROM tb_mensagem") or die("ERRO no SQL :
".mysql_error());
...
?>
```

Não deixe de nos enviar críticas ou sugestões para o próximo assunto, afinal a coluna é de vocês.

Autor/fonte: Júlio César Martini
E-mail/Url: n/a
Site: http://www.htmlstaff.org

Encode e Decode usando base64

$str = 'Ribamar Ferreira de Sousa';
//echo base64_encode($str);

$str2 = 'UmliYW1hciBGZXJyZWlyYSBkZSBTb3VzYQ== ';
echo base64_decode($str2);

Encode-Decode in PHP

```
<?php

/* Tutorial by AwesomePHP.com -> www.AwesomePHP.com */
/* Function: Encode or Decode anything based on a string */

$secretPass = 'kljhflk73#OO#*U$O(*YO';
$encodeThis = 'Please meet me at 05:44 time.';

/* Regular Encoding */
$encoded = Encode($encodeThis,$secretPass);
/* Another pass to decode */
$decoded = Encode($encoded,$secretPass);

echo 'Encoded String: '.$encoded;
echo '<br />Decoded String: '.$decoded;

/* Important: If passing this value via URL you might want to make it
explorer friendler */
$encoded = bin2hex(Encode($encodeThis,$secretPass));
/* Another pass to decode */
$decoded = Encode(hex2bin($encoded),$secretPass);

echo '<br /><br />Encoded String: '.$encoded;
echo '<br />Decoded String: '.$decoded;

function Encode($data,$pwd)
{
    $pwd_length = strlen($pwd);
    for ($i = 0; $i < 255; $i++) {
        $key[$i] = ord(substr($pwd, ($i % $pwd_length)+1, 1));
        $counter[$i] = $i;
    }
    for ($i = 0; $i < 255; $i++) {
        $x = ($x + $counter[$i] + $key[$i]) % 256;
        $temp_swap = $counter[$i];
        $counter[$i] = $counter[$x];
        $counter[$x] = $temp_swap;
    }
    for ($i = 0; $i < strlen($data); $i++) {
        $a = ($a + 1) % 256;
        $j = ($j + $counter[$a]) % 256;
        $temp = $counter[$a];
```

```
        $counter[$a] = $counter[$j];
        $counter[$j] = $temp;
        $k = $counter[(($counter[$a] + $counter[$j]) % 256)];
        $Zcipher = ord(substr($data, $i, 1)) ^ $k;
        $Zcrypt .= chr($Zcipher);
    }
    return $Zcrypt;
}

function hex2bin($hexdata) {
    for ($i=0;$i<strlen($hexdata);$i+=2) {
        $bindata.=chr(hexdec(substr($hexdata,$i,2)));
    }
    return $bindata;
}
?>
```

Site de origem: http://www.awesomephp.com/?Tutorials*3/Encode-Decode-in-PHP.html

```
<?php
Encriptar
$cod = base64_encode("senha");

print $cod;

Decriptar
echo "<br>".base64_decode($cod);

        /**
         * Decodes a text with many algorithms many times
         *
         * @param   string   $text    the text to decode
         * @param   string   $mode  the sequence of decryption methods separated by commas [,]
         *                   G      : gEncryption method using the class gEncrypter
         *                   base64 : base64 decoding algorithm
         *                   uu     : decryption using the function uudecode()
         *                 [default: G]
         * @param   string   $key    the key to use separated by commas [,]
         *                 one for each "G" you put in the sequence
         *                 [gEncrypter only] [default: ilarvet]
         * @return   string        the text finally decrypted
         */
        function decode($text, $mode = "G", $key = "ilarvet")
        {
           $exmode = explode(",", $mode);
           $exkey = explode(",", $key);

           $kcount = 0;
           $dec = $text;

           for ($i=0; $i<count($exmode); $i++)
```

```
    {
        $exmode[$i] = trim($exmode[$i]);

        switch (strtolower($exmode[$i]))
        {
            case "g":
                include_once($this->include_path . "gEncrypter.php");
                $e = new gEncrypter;
                $dec = $e->gED($dec, $exkey[$kcount]);
                $kcount ++;
                break;

            case "base64":
                $dec = base64_decode($dec);
                break;

            case "uu":
                $dec = convert_uudecode($dec);
                break;

            default:
                break;
        }
    }

    return $dec;
}

function encode($text, $mode = "G", $key = "ilarvet")
{
    $exmode = explode(",", $mode);
    $exkey = explode(",", $key);

    $kcount = 0;
    $enc = $text;

    for ($i=0; $i<count($exmode); $i++)
    {
        $exmode[$i] = trim($exmode[$i]);

        switch (strtolower($exmode[$i]))
        {
            case "g":
                include_once($this->include_path . "gEncrypter.php");
                $e = new gEncrypter;
                $enc = $e->gED($enc, $exkey[$kcount]);
                $kcount ++;
                break;
```

```
            case "base64":
                $enc = base64_encode($enc);
                break;

            case "uu":
                $enc = convert_uuencode($enc);
                break;

            default:
                break;
        }
    }
}

?>
```

9.21 – Trabalhando com e-mails em PHP

Enviar e-mail simples:
mail("cursos@ribafs.org", "Aqui fica o assunto", "Aqui o corpo do e-mail");

Envio de e-mail com tratamento de erro:

```
$to = cursos@ribafs.org';
$subject = 'Assunto';
$from = 'elias@ribafs.org';
$message = 'Como vai?';

if(mail($to, $subject, $message, "From: $from"))
  echo "E-mail enviado com sucesso!";
else
  echo "Falha no envio do e-mail – mensagem não enviada"; ?>
```

Mais um:

```
$subject = "Assunto";
        $content = "Text type email does not need br tags for line breaks;

A simple line break in your code will do the trick, as the Content-type is set to text in the header.";

        $headers  = 'MIME-Version: 1.0' . "\n";
        $headers .= 'Content-type: text; charset=iso-8859-1' . "\n";

        $headers .= 'From: info@domain.com';
        mail($to,$subject,$content,$headers);
```

Mais um exemplo de envio de e-mail com PHP:

```
$headers = "From: Testando Envio de Email <$assunto>";

// pegando data
$date = date("d/m/Y h:i");

// teoricamente iremos enviar para esse email fictício, mas podemos trazer de um BD o email sem
algum problema.
$seuemail = "joao@testando.com.br";

// assunto do email, como vai ficar em sua caixa de entrada, cuidado para não colocar nomes
inseridas em blacklists de email.
$assunto = "Sistema de envio de Email";


// Corpo do texto, fiz um exemplo básico de cadastro trazendo em variavel.
$mensagem = "
                Nome: $variavel1
                Endereco: $variavel2
                Bairro: $variavel3
```

```
                Telefone: $variavel4
        Email: $variavel5

        Enviado em: $date";


// eis aqui o envio própriamente dito.... essa linha é onde se dispara o email.
mail($seuemail, $assunto, $mensagem, $headers);
```

Envio de E-mail no formato HTML

```
<?php
//define the receiver of the email
$to = 'youraddress@example.com';
//define the subject of the email
$subject = 'Test HTML email';
//create a boundary string. It must be unique
//so we use the MD5 algorithm to generate a random hash
$random_hash = md5(date('r', time()));
//define the headers we want passed. Note that they are separated with \r\n
$headers = "From: webmaster@example.com\r\nReply-To: webmaster@example.com";
//add boundary string and mime type specification
$headers .= "\r\nContent-Type: multipart/alternative; boundary=\"PHP-alt-".$random_hash."\"";
//define the body of the message.
ob_start(); //Turn on output buffering
?>
--PHP-alt-<?php echo $random_hash; ?>
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

Hello World!!!
This is simple text email message.

--PHP-alt-<?php echo $random_hash; ?>
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

<h2>Hello World!</h2>
<p>This is something with <b>HTML</b> formatting.</p>

--PHP-alt-<?php echo $random_hash; ?>--
<?php
//copy current buffer contents into $message variable and delete current output buffer
$message = ob_get_clean();
//send the email
$mail_sent = @mail( $to, $subject, $message, $headers );
//if the message is sent successfully print "Mail sent". Otherwise print "Mail failed"
echo $mail_sent ? "Mail sent" : "Mail failed";
```

```
?>
```

Envio de e-mail com anexo:

```
<?php
//define the receiver of the email
$to = 'youraddress@example.com';
//define the subject of the email
$subject = 'Test email with attachment';
//create a boundary string. It must be unique
//so we use the MD5 algorithm to generate a random hash
$random_hash = md5(date('r', time()));
//define the headers we want passed. Note that they are separated with \r\n
$headers = "From: webmaster@example.com\r\nReply-To: webmaster@example.com";
//add boundary string and mime type specification
$headers .= "\r\nContent-Type: multipart/mixed; boundary=\"PHP-mixed-".$random_hash."\"";
//read the atachment file contents into a string,
//encode it with MIME base64,
//and split it into smaller chunks
$attachment = chunk_split(base64_encode(file_get_contents('attachment.zip')));
//define the body of the message.
ob_start(); //Turn on output buffering
?>
--PHP-mixed-<?php echo $random_hash; ?>
Content-Type: multipart/alternative; boundary="PHP-alt-<?php echo $random_hash; ?>"

--PHP-alt-<?php echo $random_hash; ?>
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

Hello World!!!
This is simple text email message.

--PHP-alt-<?php echo $random_hash; ?>
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

<h2>Hello World!</h2>
<p>This is something with <b>HTML</b> formatting.</p>

--PHP-alt-<?php echo $random_hash; ?>--

--PHP-mixed-<?php echo $random_hash; ?>
Content-Type: application/zip; name="attachment.zip"
Content-Transfer-Encoding: base64
Content-Disposition: attachment

<?php echo $attachment; ?>
--PHP-mixed-<?php echo $random_hash; ?>--
```

```
<?php
//copy current buffer contents into $message variable and delete current output buffer
$message = ob_get_clean();
//send the email
$mail_sent = @mail( $to, $subject, $message, $headers );
//if the message is sent successfully print "Mail sent". Otherwise print "Mail failed"
echo $mail_sent ? "Mail sent" : "Mail failed";
?>
```

9.22 – Trabalhando com Data e Hora em PHP

```
<?php
// Funções diversas

// Definir o horário brasileiro:
//date_default_timezone_set('Brazil/East');


// Somar dias a data atual
function add_dias($dias)
{
    return date('d/m/Y',mktime(0,0,0,date('m'),date('d')+$dias,date('Y')));
}

   /*
      Função SomarDias()
      Usada para calcular a data de feriados móveis.
      Esta função adiciona 'n_dias' à data 'data', passado como argumento, a qual deve estar no
formato dd/mm/yyyy ou yyyy-mm-dd.
      O argumento 'forma' serve para especificar o formato da data retornada. Ele pode conter os
seguintes valores:

      "pt" => Retornará a data no formato DD/MM/YYYY
      "en" => Retornará a data no formato YYYY-MM-DD

      Se 'forma' não for especificada, adotar-se-á a forma brasileira (pt).
   */
   public function SomarDias ($data, $n_dias, $forma = "pt")
   {
      if (!is_int ($n_dias))
      {
         echo "<p>Função <strong>". __FUNCTION__ ."</strong>: o argumento \"n_dias\" deve ser
um número inteiro.</p>";
         return false;
      }

      $forma = strtolower ($forma);
      if ($forma != "en" AND $forma != "pt")
         $forma = "pt";

      if (preg_match ("/^[0-9]{2}\/[0-9]{2}\/[0-9]{4}$/", $data))
         list ($dia, $mês, $ano) = explode ("/", $data);
      elseif (preg_match ("/^[0-9]{4}-[0-9]{2}-[0-9]{2}$/", $data))
         list ($ano, $mês, $dia) = explode ("-", $data);
      else
      {
         echo "<p>Função <strong>". __FUNCTION__ ."</strong>: Formato de data inválido (".
```

```
$data .").</p>";
        return false;
    }

    //transforma $n_dias em segundos
    //86400 = 60 * 60 * 24
    $segs_n_dias = $n_dias * 86400;

    // tranforma $data em timestamp
    $segs_data = strtotime ($ano . "-" . $mês . "-" . $dia);
    $segs_nova_data = $segs_data + $segs_n_dias;
    $nova_data = ($forma == "pt") ? date("d/m/Y", $segs_nova_data) : date("Y-m-d",
$segs_nova_data);

    return $nova_data;
  }



  /*
    Função SubtrairDias()
    Usada para calcular a data de feriados móveis.
    Esta função subtrai 'n_dias' da data 'data', passado como argumento, a qual deve estar no
formato dd/mm/yyyy ou yyyy-mm-dd.
    O argumento 'forma' serve para especificar o formato da data retornada. Ele pode conter os
seguintes valores:

    "pt" => Retornará a data no formato DD/MM/YYYY
    "en" => Retornará a data no formato YYYY-MM-DD

    Se 'forma' não for especificada, adotar-se-á a forma brasileira (pt).
  */
  public function SubtrairDias ($data, $n_dias, $forma = "pt")
  {
    if (!is_int ($n_dias))
    {
        echo "<p>Função <strong>". __FUNCTION__ ."</strong>: O argumento \"n_dias\" deve
ser um número inteiro.</p>";
        return false;
    }

    $forma = strtolower ($forma);
    if ($forma != "en" AND $forma != "pt")
        $forma = "pt";

    if (preg_match ("/^[0-9]{2}\/[0-9]{2}\/[0-9]{4}$/", $data))
        list ($dia, $mês, $ano) = explode ("/", $data);
    elseif (preg_match ("/^[0-9]{4}-[0-9]{2}-[0-9]{2}$/", $data))
```

```
        list ($ano, $mês, $dia) = explode ("-", $data);
      else
      {
        echo "<p>Função <strong>". __FUNCTION__ ."</strong>: Formato de data inválido (".
$data .").</p>";
        return false;
      }

      //transforma $n_dias em segundos
      //86400 = 60 * 60 * 24
      $segs_n_dias = $n_dias * 86400;

      // tranforma $data em timestamp
      $segs_data = strtotime ($ano . "-" . $mês . "-" . $dia);
      $segs_nova_data = $segs_data - $segs_n_dias;
      $nova_data = ($forma == "pt") ? date("d/m/Y", $segs_nova_data) : date("Y-m-d",
$segs_nova_data);

      return $nova_data;
   }

// Somar meses
function add_meses($meses)
{
    return date('d/m/Y',mktime(0,0,0,date('m')+$meses,date('d'),date('Y')));
}

// Somar anos
function add_anos($anos)
{
    return date('d/m/Y',mktime(0,0,0,date('m'),date('d'),date('Y')+$anos));
}

function somar_dias_uteis($str_data,$int_qtd_dias_somar = 7) {
   // Caso seja informado uma data do MySQL ou PostgreSQL do tipo DATETIME - aaaa-mm-dd
00:00:00
   // Transforma para DATE - aaaa-mm-dd

   $str_data = substr($str_data,0,10);
   // Se a data estiver no formato brasileiro: dd/mm/aaaa
   // Converte-a para o padrão americano: aaaa-mm-dd
   if ( preg_match("@/@",$str_data) == 1 ) {
      $str_data = implode("-", array_reverse(explode("/",$str_data)));
   }
   $array_data = explode('-', $str_data);
   $count_days = 0;
   $int_qtd_dias_uteis = 0;
   while ( $int_qtd_dias_uteis < $int_qtd_dias_somar ) {
```

```
        $count_days++;
            if ( ( $dias_da_semana = gmdate('w', strtotime('+'.$count_days.' day', mktime(0, 0, 0,
$array_data[1], $array_data[2], $array_data[0]))) ) != '0' && $dias_da_semana != '6' ) {
          $int_qtd_dias_uteis++;
        }
    }
    return gmdate('d/m/Y',strtotime('+'.$count_days.' day',strtotime($str_data)));
}

//Exemplo de uso:
//echo somar_dias_uteis('05/12/2006');
//echo somar_dias_uteis('2006-12-01',15);
// http://leandrovieira.com/archive/somando-dias-uteis-a-uma-data-especifica-com-php


function SomarData($data, $dias, $meses, $ano)
{
  //passe a data no formato dd/mm/yyyy
   $data = explode('/', $data);
   $newData = date('d/m/Y', mktime(0, 0, 0, $data[1] + $meses,
   $data[0] + $dias, $data[2] + $ano) );
   return $newData;
}

//Exemplo de como usar:
//echo SomarData('04/04/2007', 1, 2, 1);

//Este exemplo acima estamos adicionando 1 dia, 2 meses e 1 ano na data informada. O resultado
então seria '05/06/2008'
//http://www.codigofonte.com.br/codigo/php/data-hora/funcao-para-somar-datas-em-php


/*
Funções sub_data() e som_data()
Desenvolvidas por
InFog (Evaldo Junior Bento)
em Junho de 2007
junior_pd_bento@yahoo.com.br
Este script é disponibilizado utilizando
a licença GPL em sua versão mais atual.
Distribua, aprenda, ensine
mas mantenha os créditos do autor
Viva ao Software Livre e à livre informação
*/

/*
Função sub_data()
Esta função recebe a data no formato brasileiro dd/mm/AAAA
```

```
e o número de dias que serão subtraídos dela.
Certifique-se de checar se a data é válida antes de chamar a função
*/

function sub_data($data, $dias) {
    $data_e = explode("/",$data);
    $data2 = date("m/d/Y", mktime(0,0,0,$data_e[1],$data_e[0] - $dias,$data_e[2]));
    $data2_e = explode("/",$data2);
    $data_final = $data2_e[1] . "/". $data2_e[0] . "/" . $data2_e[2];
    return $data_final;
}

/*
Função soma_data()
Esta função recebe a data no formato brasileiro dd/mm/AAAA
e o número de dias que serão adicionados à dela.
Certifique-se de checar se a data é válida antes de chamar a função
*/

function soma_data($data, $dias) {
    $data_e = explode("/",$data);
    $data2 = date("m/d/Y", mktime(0,0,0,$data_e[1],$data_e[0] + $dias,$data_e[2]));
    $data2_e = explode("/",$data2);
    $data_final = $data2_e[1] . "/". $data2_e[0] . "/" . $data2_e[2];
    return $data_final;
}


//Autor/fonte: Evaldo Junior
//E-mail/Url: http://tuxmasters.blogspot.com/2007/06/somando-datas-no-php.html

function hoje(){
    return date("d/m/Y");
}

function agora(){
    return date("H:i:s");
}

function databr_atual(){
    return date("d/m/Y");
}

function hora_atual(){
    return date("H:i:s");
}
```

```
//Verifica se Data1 é menor que Data2, além de validá-las
//Entrada no formato Brasileiro
function ChecaVarData($data1,$data2)
{
 if (DataValida($data1) and DataValida($data1))
 {
  $data1=DataBrasilMySQL($data1);
  $data2=DataBrasilMySQL($data2);
  Return($data1<=$data2);
 }
 else
 {
  Return(false);
 }

}

function data_valida($data){
	$d = explode("/",$data);
	$d2 = checkdate ($d[1],$d[0],$d[2]);
	if (!$d2){
		?><script>alert("A data "+<?=$d[0].$d[1].$d[2] ?>+" não é válida!")</script><?php
		exit;
	}
}

function idade($dia,$mes,$ano) {
	if($mes >= date(m)) {
		if($dia >=date(d)) {
			$idade = date(Y) - $ano + 1;
		}else{
			$idade = date(Y) - $ano;
		}
	}else{
		$idade = date(Y) - $ano;
	}
	return $idade;
}
//echo idade(03,08,1956);

/*
 *
 *
$arrDia=array ('Domingo','Segunda','Terça','Quarta','Quinta','Sexta','Sábado ');
$arrMes=array (1=>'Janeiro','Fevereiro','Março','Abril','Maio','Junho',
'Julho','Agosto','Setembro','Outubro','Novembro','Dezembro');
?>
<form>
```

```
Selecione a data completa<br>
<select name="semana">
<?php
for($i = 0; $i < 7; $i++){
echo"<option> $arrDia[$i]";
}
echo'</select><select name="dia">';
for ($i=1;$i<=31; $i++){
echo"<option>$i";
}
echo '</select> de <select name="mes">';
for ($i=1;$i<=12;$i++){
echo "<option value=\"$i\"> $arrMes[$i]";
}
echo '</select> de <select name="ano">';  //Faltava ;
$start_year = date('Y') - 10;
$end_year = $start_year + 20;
for($i = $start_year; $i <= $end_year; $i++){
echo "<option> $i";
}

 */

$arrDia=array
('Domingo','Segunda','Terça','Quarta','Quinta','Sexta','Sábado
');
$arrMes=array
(1=>'Janeiro','Fevereiro','Março','Abril','Maio','Junho',
'Julho','Agosto','Setembro','Outubro','Novembro','Dezembro');
?>
<form>
Selecione a data completa<br>
<select name="semana">
<?php
for($i = 0; $i < 7; $i++){
echo"<option> $arrDia[$i]";
}
echo'</select><select name="dia">';
for ($i=1;$i<=31; $i++){
echo"<option>$i";
}
echo '</select> de <select name="mes">';
for ($i=1;$i<=12;$i++){
echo "<option value=\"$i\"> $arrMes[$i]";
}
echo '</select> de <select name="ano">';  //Faltava ;
$start_year = date('Y') - 10;
$end_year = $start_year + 20;
```

```
for($i = $start_year; $i <= $end_year; $i++){
echo "<option> $i";
}


/* Código Original/ Original Code by Woodys
* http://www.zend.com/codex.php?id=176
*
* adaptação/tradução para o português e formato brasileiro das datas
*
* Alguns exclarecimentos
* - O que é TIMESTAMP do Unix?
* - É a contagem, em segundos, desde o dia 1 de janeiro de 1970 00:00:00 GMT
*     ,i.e., os segundos que se passaram até momento desde as ZERO horas do dia 1 de janeiro de
1970
*     exemplo:
*     timestamp = 1042752929  => passaram-se 1042752929 segundos desde 1/jan/1970 00horas
00min 00 seg
*
*  Tradução e Adaptação by Calvin
*/

?>

<?php

/**
* Calcula a quantidade de dias úteis entre duas datas (sem contar feriados)
* @author Marcos Regis
* @param String $datainicial
* @param String $datafinal=null
*/
function dias_uteis($datainicial,$datafinal=null){
  if (!isset($datainicial)) return false;
  if (!isset($datafinal)) $datafinal=time();

$segundos_datainicial = strtotime(preg_replace("#(\d{2})/(\d{2})/(\d{4})#","$3/$2/$1",
$datainicial));
$segundos_datafinal = strtotime(preg_replace("#(\d{2})/(\d{2})/(\d{4})#","$3/$2/$1",$datafinal));
$dias = abs(floor(floor(($segundos_datafinal-$segundos_datainicial)/3600)/24 ) );
$uteis=0;
for($i=1;$i<=$dias;$i++){
$diai = $segundos_datainicial+($i*3600*24);
$w = date('w',$diai);
if ($w==0){
//echo date('d/m/Y',$diai)." é Domingo<br />";
}elseif($w==6){
//echo date('d/m/Y',$diai)." é Sábado<br />";
```

```
}else{
//echo date('d/m/Y',$diai)." é dia útil<br />";
$uteis++;
}
}
return $uteis;
}
?>
```

ex. de uso

```
<?php
$data='28/02/2007';
echo "Existem ".dias_uteis($data)." dias úteis entre $data e hoje";
?>
```

```
<?php

//Impresso através do site http://www.htmlstaff.org

//Função para pegar todos os feriados do ano

//Uma função para pegar todos os feriados do ano (fixos e móveis). Coloquei dois estaduais, mas
não será problema em editá-los conforme a cidade/estado.

//Função:

/**
* @param $ano ano em que se quer calcular os feriados
* @return array com os feriados do ano (fixo e moveis)
*/
function getFeriados($ano){
	$dia = 86400;
	$datas = array();
	$datas['pascoa'] = easter_date($ano);
	$datas['sexta_santa'] = $datas['pascoa'] - (2 * $dia);
	$datas['carnaval'] = $datas['pascoa'] - (47 * $dia);
	$datas['corpus_cristi'] = $datas['pascoa'] + (60 * $dia);
	$feriados = array (
		'01/01', // Confraternização universal
		'02/02', // Navegantes
		date('d/m',$datas['carnaval']),
		date('d/m',$datas['sexta_santa']),
		date('d/m',$datas['pascoa']),
		'21/04', //Tiradentes
		'01/05',
		date('d/m',$datas['corpus_cristi']),
		'20/09', // Revolução Farroupilha m/
		'12/10',
		'02/11',
```

```
            '15/11',
            '25/12',
            );
    return $feriados;
}
$feriados = getFeriados('2008');

for($x=0;$x<12;$x++){
    print $feriados[$x].'<br>';
}

//Autor/fonte: Maiquel Leonel
//E-mail/Url: http://www.vivaolinux.com.br/scripts/verScript.php?codigo=3430
?>
```

Feriados Brasileiros

Feriados nacionais

Os feriados nacionais são definidos pelas seguintes leis: nº 662 (de 1949)[1], nº 6.802 (de 1980)[2], nº 9.093 (de 1995)[3], e nº 10.607 (de 2002)[4]. Os feriados nacionais brasileiros são:

Data	Feriado	Motivação
1º de janeiro	Confraternização Universal	social
21 de abril	Tiradentes	cívica
1º de maio	Dia do Trabalho	social
7 de setembro	Independência do Brasil	cívica
12 de outubro	Nossa Senhora Aparecida	religiosa (católica)
2 de novembro	Finados	religiosa (católica)
15 de novembro	Proclamação da República	cívica
25 de dezembro	Natal	religiosa (cristã)

Feriados móveis

São feriados que dependem da Páscoa. Os judeus celebram a Páscoa segundo o que prescreve o livro do Êxodo, no capítulo 12, no dia 14 do mês de Nissan. É a celebração da libertação da escravidão do Egito para a liberdade da Terra Prometida por Deus a Abraão. O cristianismo celebra a Páscoa cristã, Ressurreição de Cristo, acompanhando de certa forma a data Páscoa judaica. Mas o calendário judeu é baseado na Lua, então a data da Páscoa cristã passou a ser móvel no calendário cristão, assim como as demais datas referentes à Páscoa, tanto na Igreja Católica como nas Igrejas Protestantes e Igrejas Ortodoxas. O primeiro Concilio geral da Igreja, o de Nicéia, no ano 325, determinou que a Páscoa cristã seria celebrada no domingo seguinte à primeira Lua cheia após o equinócio da primavera do hemisfério Norte (21 de março); podendo ocorrer entre 22 de março e 25 de abril.

Outros feriados que dependem da Páscoa:

Carnaval (terça-feira) - quarenta e sete dias antes da Páscoa.

Quaresma - inicia na quarta-feira de cinzas e termina no Domingo de Ramos (uma semana antes da Páscoa).

Sexta-feira Santa - a sexta-feira imediatamente anterior Sábado da Solene Vigília Pascal - o sábado de véspera

Pentecostes - o oitavo domingo após a Páscoa.

Corpus Christi - a quinta-feira imediatamente após o Pentecostes.

Feriados estaduais

A Lei nº 9.093, de 12 de setembro de 1995, incluiu entre os feriados civis, antes apenas os declarados em lei federal, a "data magna do Estado fixada em lei estadual".

Acre

Data	Feriado
23 de janeiro	Dia do evangélico
15 de junho	Aniversário do estado
6 de agosto	Início da Revolução Acreana
5 de setembro	Dia da Amazônia
17 de novembro	Assinatura do Tratado de Petrópolis

Alagoas

Data	Feriado
24 de junho	São João (decreto estadual n.4.087, de 18 de dezembro de 2008)
29 de junho	São Pedro (decreto estadual 4.087, de 18 de dezembro de 2008)
16 de setembro	Emancipação política

20 de novembro Dia da Consciência Negra

Amapá
Data Feriado
19 de março Dia de São José
5 de outubro Criação do estado
20 de novembro Dia da Consciência Negra

Amazonas
Data Feriado
5 de setembro Elevação do Amazonas à categoria de província
20 de novembro Dia da Consciência negra
8 de dezembro Dia de Nossa Senhora da Conceição

Bahia
Data Feriado
2 de julho Independência da Bahia
20 de novembro Dia da Consciência Negra

Ceará

O Ceará não tem data magna. Em 2007, o deputado estadual Lula Morais apresentou o Projeto de Lei nº 53, para definir o dia 19 de março como Data Magna do Estado do Ceará. O relator do projeto na Comissão de Constituição e Justiça da Assembleia Legislativa, deputado João Jaime, devolveu o projeto ao autor, com a sugestão de transformá-lo em projeto de indicação. Em junho de 2007, o projeto de lei foi retirado pelo autor e arquivado. Desde então, ninguém mais apresentou projeto que fixasse a data magna estadual cearense.

Distrito Federal
Data Feriado
21 de abril Fundação de Brasília
30 de novembro Dia do Evangélico

Espírito Santo
Data Feriado
23 de maio Colonização do solo espírito-santense
28 de outubro Dia do Servidor Público

Goiás
Data Feriado
28 de outubro Dia do Servidor Público

Maranhão
Data Feriado
28 de julho Adesão do Maranhão à independência do Brasil
8 de dezembro Dia de Nossa Senhora da Conceição

Mato Grosso
Data Feriado
20 de novembro Dia da Consciência Negra

Mato Grosso do Sul
Data Feriado
11 de outubro Criação do estado

Minas Gerais
Data Feriado
21 de abril Tiradentes

Pará
Data Feriado
15 de agosto Adesão do Grão-Pará à independência do Brasil
8 de dezembro Nossa Senhora da Conceição

Paraíba
Data Feriado
5 de agosto Emancipação política do estado

Paraná
Data Feriado
19 de dezembro Emancipação política (emancipação do Paraná de São Paulo)

Pernambuco
Data Feriado Legislação Observações
6 de março Revolução Pernambucana de 1817 Lei nº 13.386[1], de 24 de dezembro de 2007 ponto facultativo

Piauí
Data Feriado
13 de março Dia da Batalha do Jenipapo
19 de outubro Dia de Piauí

Rio de Janeiro
Data Feriado
23 de abril Dia de São Jorge
15 de outubro Dia do Comércio
20 de novembro Dia da Consciência Negra

Rio Grande do Norte
Data Feriado
29 de junho Dia de São Pedro
3 de outubro Mártires de Cunhaú e Uruaçu

Rio Grande do Sul
Data Feriado

20 de setembro Revolução Farroupilha

Rondônia
Data Feriado
4 de janeiro Criação do estado
18 de junho Dia do Evangélico

Roraima
Data Feriado
5 de outubro Criação do estado

Santa Catarina
Data Feriado
11 de agosto Criação da capitania, separando-se de São Paulo

São Paulo
Data Feriado
25 de janeiro Dia de São Paulo
9 de julho Revolução Constitucionalista de 1932
20 de novembro Dia da Consciência Negra
12 de outubro Nossa Senhora da Conceição Aparecida

Sergipe
Data Feriado
8 de julho Autonomia política de Sergipe

Tocantins
Data Feriado
5 de outubro Criação do estado

Feriados municipais

Os municípios podem declarar, em lei municipal, até quatro feriados religiosos, de acordo com a tradição local, entre eles a Sexta-Feira da Paixão. A Lei nº 9.335, de 10 de dezembro de 1996[7], acrescentou, ainda, como feriado civil, os dias do início e do término do ano do centenário de fundação do município, desde que fixado em lei municipal.

Fonte: http://pt.wikipedia.org/wiki/Feriados_no_Brasil

Referências

http://php.net
http://pt.wikibooks.org/wiki/Aplicativos_em_PHP

Desenvolvimento web:
PHP Orientado à Objetos

Lucas Francisco da Matta Vegi
www.mr-bin.blogspot.com

Público-Alvo

- Alunos de cursos voltados para tecnologia:
  - Sistemas de Informação
  - Ciência da Computação
  - etc...

- Pessoas que já possuem conhecimento prévio de lógica de programação e HTML.

Objetivo

- O objetivo desse curso é ensinar os conceitos básicos da linguagem PHP, proporcionando aos alunos conhecimento suficiente para iniciarem seus projetos de programação orientada à objeto utilizando está linguagem.

- Com a base adquirida nesse curso, a aprendizagem autodidata de outros recursos PHP se tornará muito mais fácil.

Leituras Recomendadas

Desenvolvimento web: PHP Orientado à Objeto
Lucas Francisco da Matta Vegi
www.mr-bin.blogspot.com

Conteúdo Programático

- Módulo I -
Introdução ao Mundo PHP

- Módulo II -
A Linguagem PHP

- Módulo III -
PHP OO - Programação Orientada à Objetos com PHP

- Módulo IV -
Solução para os dados voláteis: Sessão

- Módulo I -
Introdução ao Mundo PHP

Desenvolvimento web:
PHP Orientado à Objetos

Lucas Francisco da Matta Vegi
www.mr-bin.blogspot.com

Vida de programador!

http://www.youtube.com/watch?v=apREUmNp9Ec&feature=related

Alguém se identificou???

Após o momento de descontração,
agora vamos trabalhar!

Vou aprender a pensar no curso?

Não!..Você não vai aprender a pensar aqui!

- Durante o curso, você vai aprender a sintaxe do PHP e a lógica dos recursos que ele oferece, **porém isso não garante que você irá fazer bons softwares** com esse conhecimento.

- Todos nós sabemos a gramática da língua portuguesa e como se escreve, porém nem todos nós temos o dom de escrever boas redações. **Com linguagens de programação é a mesma coisa!**

- No português, a prática da escrita tráz melhoras consideráveis na qualidade das redações escritas por quem exercita. A prática da programação também tráz melhoras de raciocínio e qualidade de software dos programadores!

O que é Programação Server-Side?

- PHP é uma linguagem de programação interpretada pelos servidores na internet (server-side).
- O Servidor recebe as requisições dos clientes, processa os scripts PHP e retorna para os clientes o PHP já processado em forma de HTML que é visualizado no navegador(browser).
- O cliente **NUNCA** conseguirá ver o código PHP.

Demonstrando programa Server-Side

Isso te faz lembrar alguma coisa?

PHP é computação nas nuvens!

- Todo sistema desenvolvido em PHP se enquadra no novo conceito de computação nas nuvens.
- Nesse novo conceito computacional, os dados e softwares migram dos computadores e servidores locais para as nuvens de informação, que são milhares de servidores em cluster espalhados por diversos pontos geográficos.
- Algumas das vantagens da utilização do PHP se confundem com as vantagens da computação nas nuvens.

Vantagens da utilização do PHP

- PHP é software livre!
- Portabilidade (Independe de Sistema Operacional ou equipamento)
- Mobilidade (Acesso aos dados de qualquer lugar!)
- Os softwares desenvolvidos são leves na visão do cliente.
- Possui uma das maiores comunidades de programadores da internet:
  - iMasters - http://forum.imasters.uol.com.br/
  - HTMLSTAFF - http://www.htmlstaff.org/
  - **PHP MG...**

Grupo de desenvolvedores PHP de Minas Gerais

- ***Site do Grupo:*** http://www.phpmg.com
- ***Lista do Google Groups:*** http://groups.google.com/group/phpmg
- ***Comunidade do Orkut:*** http://www.orkut.com.br/Main#Community.aspx?cmm=26992151

Ambiente WAMP

- Para programar em PHP no ambiente Windows, as tecnologias mais utilizadas atualmente e que precisam estarem instaladas no computador do programador são:
  - **W**indows
  - **A**pache
  - **M**ySQL
  - **P**HP
- WAMP é um sistema indicado para os usuários que não têm instalado no sistema nenhuma dessas tecnologias (o Windows já deve estar instalado!), já que ele realiza uma instalação completa e desde o zero.
  - ***DOWNLOAD:*** http://www.wampserver.com/en/download.php
- Todos os seus programas PHP devem ser salvos na pasta chamada "**WWW**" do WAMP.

Ferramentas para desenvolvimento

- ***NotePad***

- ***NotePad++ :*** http://www.baixaki.com.br/download/notepad-.htm

- ***Eclipse:*** http://www.eclipse.org/pdt/

- ***Zend Studio:*** http://www.zend.com/en/products/studio/downloads

Hello World!

Hello World!

```
<?php
    echo("<center>Hello World!</center>");
?>
```

Cadê o PHP??

- Já no navegador visualizando o seu Hello World!, entre na opção "exibir código-fonte" (Ctrl+U no Firefox).

- Você deve ter reparado que só apareceu códigos HTML. Isso acontece porque o servidor já processou o código PHP quando você fez a requisição de abrir a página e devolveu ao cliente (browser) somente o HTML de visualização.

- Módulo II -

A Linguagem PHP

Desenvolvimento web:

PHP Orientado à Objetos

Lucas Francisco da Matta Vegi
www.mr-bin.blogspot.com

PHP escreve HTML!

Conceitos básicos da Linguagem

- Todo programador PHP deve conhecer muito bem as Tags HTML, pois todas as partes visuais das aplicações PHP são feitas pelo HTML.

- O PHP escreve HTML e também pode ser escrito misturado com o HTML.

- Como o PHP tem a capacidade de escrever HTML, ele pode ser usado para "formatar" a apresentação do PHP, dando cores, formas e posições na tela.

  - ***Exemplo prático:*** Desenvolver uma página que escreve uma tabela HTML utilizando o PHP.

Exemplo Tabela HTML com PHP

```
<?php  //arquivo "tabela_com_php.php"  ?>
<html>
<head><title>Tabela HTML com PHP</title></head>
<body>
<?php
   echo("<table align='center' border='1'>
         <tr>
            <th colspan='2' align='center'>Tabela HTML com PHP</th>
         </tr>
         <tr>
            <td align='center'>Lucas</td><td align='center'>Vegi</td>
         </tr>");
?>
</body>
</html>
```

Tipos de dados

- PHP suporta vários tipos de dados, dentre eles os principais são:
  - **Boolean**
  - **Integer**
  - **Float**
  - **String**
  - **Array**
  - **Objeto (Veremos no módulo III)**

- A tipagem em PHP é dinâmica, ou seja, em PHP **não é necessário especificar os tipos das variáveis**. É possível atribuir qualquer valor a qualquer variável, pois o interpretador PHP faz a alteração do tipo da variável de forma automática e transparente.

- Em PHP todas as variáveis são antecedidas de **$** (cifrão ou cash).

Variáveis em PHP

- Exemplo de declaração de variável em PHP:
  - **$nome_da_variavel;**

- No ***exemplo prático*** a seguir, veremos o uso de variáveis com o PHP fazendo a interpretação automática de tipos.

Exemplo das variáveis dinâmicas

```
<?php
   $teste;                                          //declara a variável

   $teste = 1;                                     //atribui um valor integer para ela
   echo("Valor inteiro: ".$teste."<br>");          //imprime o valor integer

   $teste = $teste + 3.6;                           //atribui valor float
   echo("Valor float: ".$teste."<br>");            //imprime o valor float

   $teste = true; //atribui valor boolean
   echo("Valor Boolean: ".$teste."<br>");           //imprime valor boolean

   $teste = "Agora eu sou uma String!";            //atribui valor string
   echo("Valor String: ".$teste."<br>");            //imprime valor string
?>
```

Entrada e saída de dados

- Toda entrada de dados em PHP ocorre a partir de formulários HTML.

- Esses formulários podem utilizar dois métodos de envio de dados:
  - POST
  - GET

- Toda saída de dados em PHP, como já visto anteriormente, é feita a partir da função **echo();**

  - ***Exemplo prático:*** Formulário HTML enviando dados por POST e GET

Exemplo formulário HTML com POST

```
<html><!-- Nome do arquivo exemplo_form.php -->
<head><title>Exemplo com formulários</title></head>
<body>
   <form action="exemplo_form.php" method="post" name="exemplo">
      Nome:        <input type="text" name="nome" size="30"><br>
      Mensagem: <input type="text" name="mensagem" size="30"><br>
                   <input type="submit" name="enviar" value="Enviar"><br>
   </form>
<?php
   echo("Nome: ".$_POST[nome]."<br>");
   echo("Mensagem: ".$_POST[mensagem]."<br>");
?>
</body>
</html>
```

Exemplo formulário HTML com GET

```
<html><!-- Nome do arquivo exemplo_form.php -->
<head><title>Exemplo com formulários</title></head>
<body>
   <form action="exemplo_form.php" method="get" name="exemplo">
      Nome:        <input type="text" name="nome" size="30"><br>
      Mensagem: <input type="text" name="mensagem" size="30"><br>
                   <input type="submit" name="enviar" value="Enviar"><br>
   </form>
<?php
   echo("Nome: ".$_GET[nome]."<br>");
   echo("Mensagem: ".$_GET[mensagem]."<br>");
?>
</body>
</html>
```

GET x POST

- Basicamente a maior diferença entre POST e GET, é o fato de que o método POST suporta enviar uma ***maior*** quantidade de dados e de forma ***transparente*** ao usuário.

- Já o método GET suporta enviar uma ***menor*** quantidade de dados e os envia de uma forma ***visível*** ao usuário.

Array em PHP

- Um array pode ser iniciado em PHP da seguinte forma:
  - $vetor = array(17,53,89,5);

- A partir do exemplo acima, se fosse feito um **echo($vetor[1]);** o valor impresso na tela seria **53**.

- Arrays também podém receber valores sem necessáriamente terem sido iniciados antes. Ex:
  - $vet[0] = "Valor atribuido";
  - $vet[10] = "Outro valor atribuido";

- Uma diferença dos arrays em PHP com relação a outras linguagens de programação são os **Arrays Associativos**.

Arrays Associativos

- A maior diferença entre os arrays associativos e os arrays tradicionais, é que os índices dos arrays associativos podem podem ser valores não númericos escolhidos pelo programador.

- Declaração de array associativo:
  - $vet_notas = array("joão" => 8,"maria" => 9,"josé" => 6);

- Considerando o array associativo do exemplo acima, se fosse feito um **echo($vet_notas["maria"]);** o valor impresso na tela seria **9**.

Palavras reservadas em PHP

and	or	xor	__FILE__
__LINE__	array()	as	break
class	const	continue	declare
die()	do	echo()	else
empty()	enddeclare	endfor	endif
endswitch	endwhile	eval()	extends
for	foreach	function	global
include()	include_once()	isset()	list()
print()	require()	require_once()	return()
switch	unset()	use	var
__FUNCTION__	__CLASS__	_METHOD_	final (no PHP 5)
interfaca(no PHP 5)	implements (no PHP 5)	instanceof(no PHP 5)	public (no PHP 5)
protected (no PHP 5)	abstract (no PHP 5)	clone (no PHP 5)	try (no PHP 5)
throw (no PHP 5)	cfunction (no PHP 4 somente)	old_function (no PHP 4 somente)	this (no PHP 5)
__NAMESPACE__ (no PHP 5.3)	namespace (no PHP 5.3)	goto (no PHP 6 somente)	__DIR__ (no PHP 5.3)
exception (no PHP 5)	if	catch (no PHP 5)	endforeach
case	new	final (no PHP 5)	exit()
default	static	php_user_filter(no PHP 5)	
elseif	while	private (no PHP 5)	

Operadores aritiméticos

+ adição
- subtração
* multiplicação
/ divisão
% módulo
= atribuição

Operadores relacionais e lógicos

Relacionais	Lógicos
< menor que	&& and e (conjunção)
<= menor ou igual a	|| or ou (disjunção)
> maior que	xor ou exclusivo
>= maior ou igual a	! não (negação)
== igual a	
!= diferente de	

Operadores de atribuição

+= adição
-= subtração
*= multiplicação
/= divisão
%= módulo
.= concatenação
= atribuição

- $c = $c + 3; *é equivalente a* $c += 3;
- $c = $c - 6; *é equivalente a* $c -= 6;
- $c = $c * 3; *é equivalente a* $c *= 3;
- $c = $c / 5; *é equivalente a* $c /= 5;
- $c = $c % 2; *é equivalente a* $c %= 2;
- $c = $c ."teste"; *é equivalente a* $c .= "teste";

Procedência de operadores

Operador	Descrição
(mais alta)	
! + -	não lógico, mais unário, menos unário
* / %	multiplicação, divisão, módulo
+ -	adição, subtração
< <= >= >	desigualdades
== !=	igual, diferente
&& and	e lógico
|| or xor	ou lógico , ou exclusivo
=	atribuição
(mais baixa)	

Incremento e decremento

- Assim como na linguagem C/C++ e Java, os incrementos e decrementos em PHP são feitos pelos operadores **++** e **--** respectivamente.
  - **Exemplo:**

```
<?php
    $c = 6;
    $b = $c++;
    echo($b); //imprime 6
    echo($c); //imprime 7
?>
```

  - **Exemplo 2:**

```
<?php
    $c = 6;
    $b = --$c;
    echo($b); //imprime 5
    echo($c); //imprime 5
?>
```

Estruturas de controle de fluxo

- As estruturas de controle de fluxo do PHP são **idênticas às do C/C++.**

- Em nosso curso serão estudadas as estruturas mais usadas do PHP, embora essas não sejam as únicas existentes:
  - ***Estruturas de seleção***
    - if
    - switch
  - ***Estruturas de repetição***
    - while
    - for

Estrutura de seleção - IF

```
<?php
   if( condição  )
   {
      //códigos executados quando atendida a condição
      //códigos executados quando atendida a condição
      //códigos executados quando atendida a condição
   }
   else
   {
      //códigos executados quando não atendida a condição
      //códigos executados quando não atendida a condição
   }
?>
```

- ***Exemplo prático com IF***

Exemplo comando IF

```
<html>
<head><title>Exemplo IF - Curso Desenvolvimento Web: PHP orientado à objetos</title></head>
<body>
   <form action="exemplo_if.php" method="post" name="teste_if">
      Dado 1:<input type="text" name="dado1" size="10"><br>
      Dado 2:<input type="text" name="dado2" size="10"><br>
                <input type="submit" value="Calcular">
   </form>
   <?php
      if($_POST[dado1] != "" && $_POST[dado2] != "")       //testa se o usuário digitou os dados
      {
         $total = $_POST[dado1] + $_POST[dado2];            //soma os dados digitados
         echo("O total da sua soma é: ".$total."<br>");      //imprime o total

         if($total > 15)
         {
            echo("O total da sua soma é MAIOR que 15!<br>");
         }
         else
         {
            echo("O total da sua soma é MENOR que 15!<br>");
         }
      }  //end if
   ?>
</body>
</html> <!-- nome do arquivo exemplo_if.php -->
```

Estrutura de seleção - SWITCH

```
<?php
switch ($variavel)
{
    case valor1 :
        //códigos executados quando $variavel igual valor1
        break;
    case valor2 :
        //códigos executados quando $variavel igual valor2
        break;
    default:
        //códigos executados quando $variavel diferente de valor1 e valor2
}
?>
```

- ***Exemplo prático com SWITCH***

Exemplo comando SWITCH

```
<html>
<head><title>Exemplo SWITCH - Curso Desenvolvimento Web: PHP orientado à objetos</title></head>
<body>
   <form action="exemplo_switch.php" method="post" name="teste_switch">
      Dado:<input type="text" name="dado" size="10"><br>
               <input type="submit" value="Enviar">
   </form>
   <?php
      if($_POST[dado] != "")            //testa se o usuário digitou o dado
      {
         switch($_POST[dado])       //a variável de parâmetro é o dado digitado
         {
            case "Lucas":
               echo("Dado digitado: ".$_POST[dado].". Esse cara é gente boa!<br>");
               break;

            case "PHP":
               echo("Dado digitado: ".$_POST[dado].". Essa linguagem é massa!<br>");
               break;

            default:
               echo("Dado digitado: ".$_POST[dado].". Dado não previsto!<br>");
         }//end switch
      }//end if
   ?>
</body>
</html> <!-- nome do arquivo exemplo_switch.php -->
```

Estrutura de repetição - WHILE

```
<?php
while( condição )
{
        //códigos executados enquanto atendida a condição
        //códigos executados enquanto atendida a condição
        //códigos executados enquanto atendida a condição
}
?>
```

- *Exemplo prático com WHILE*

Exemplo comando WHILE

```
<html>
<head><title>Exemplo WHILE - Curso Desenvolvimento Web: PHP orientado à
objetos</title></head>
<body>
   <form action="exemplo_while.php" method="post" name="teste_while">
      Dado:<input type="text" name="dado" size="10"><br>
            <input type="submit" value="Enviar">
   </form>
   <?php
      if($_POST[dado] != "")            //testa se o usuário digitou o dado
      {
         $aux = $_POST[dado];       //variável auxiliar recebe o valor digitado

         while($aux <= 50)
         {
            echo("Variável auxiliar: ".$aux."<br>");
            $aux += 2;                                  //soma mais 2 ao valor atual da variável auxiliar

         }//end while
      }//end if
   ?>
</body>
</html> <!-- nome do arquivo exemplo_while.php -->
```

Estrutura de repetição - FOR

```
<?php
for( inicia o contador ; condição ; incrementa ou decrementa contador )
{
    //códigos executados enquanto atendida a condição
    //códigos executados enquanto atendida a condição
    //códigos executados enquanto atendida a condição
}
?>
```

- *Exemplo prático com FOR*

Exemplo comando FOR

```
<html>
<head><title>Exemplo FOR - Curso Desenvolvimento Web: PHP orientado à objetos</title></head>
<body>
   <form action="exemplo_for.php" method="post" name="teste_for">
      Nº de linhas:<input type="text" name="linhas" size="10"><br>
                    <input type="submit" value="Enviar">
   </form>
   <?php
      if($_POST[linhas] != "")        //testa se o usuário digitou o dado
      {
         $aux = $_POST[linhas];       //variável auxiliar recebe o valor digitado
         echo("<table border='1'>
               <tr>
                  <th colspan='2' bgcolor='#bbbbbb'>Tabela Dinâmica</th>
               </tr>");                                          //inicia uma tabela HTML com cabeçalho
         for($i = 0; $i < $aux; $i++)
         {
            if($i % 2 == 0)           //testa se o valor do contador é par
            {
               $cor = "#ffffff";      //atribui valor hexadecimal de cor
            }
            else
            {
               $cor = "#bbbbbb";      //atribui valor hexadecimal de cor
            }
            echo("<tr><td bgcolor='$cor'>$i</td><td bgcolor='$cor'>$i</td></tr>");        //escreve linha da tabela com 2
células
         }//end for
         echo("</table>");     //encerra tabela HTML
      }//end if
   ?>
</body>
</html> <!-- nome do arquivo exemplo_for.php -->
```

Funções em PHP

- Assim como em praticamente todas as linguagens de programação, PHP tem suporte a criação de funções.
- Uma função é uma “caixa preta” que realiza uma tarefa sempre que ela for chamada.
- A sintaxe da criação das funções é **idêntica à do C/C++**
- Uma função pode retornar ou não valores, podendo assim ser moldada pela necessidade do programador.
- Os parâmetros das funções podem ser passados por valor ou referência.

Criação de Funções em PHP

- **Função com passagem de parâmetro por valor:**

```
function nome_da_funcao ($parametro1, $parametro2)
{
        //comandos executados pela função
        return valor_retornado ;
}
```

- **Função com passagem de parâmetro por referência:**

```
function nome_da_funcao (&$parametro1, &$parametro2)
{
        //comandos executados pela função
        return valor_retornado ;
}
```

Chamada de função em PHP

```
<?php
  function soma($a, $b)
  {
     $total = $a + $b;
     return $total;
  }

  $valor1 = 2;
  $valor2 = 3;

  $resultado = soma($valor1,$valor2);     //chama a função soma
  echo($resultado);                       //imprime 5 na tela
?>
```

Praticando funções em PHP

- Para visualizar melhor a utilização das funções em PHP, vamos fazer um ***exemplo prático*** envolvendo duas funções, uma utilizando passagem de parâmetros por valor e outra utilizando a passagem de parâmetro por referência.

Exemplo Prático com funções

Exemplo de Funções PHP

```
<html>
<head><title>Exemplo FUNÇÕES - Curso Desenvolvimento Web: PHP orientado à objetos</title></head>
<body>
   <form action="exemplo_funcoes.php" method="post" name="teste_funcoes">
      Dado:<input type="text" name="dado" size="10"><br>
               <input type="submit" value="Enviar">
   </form>
   <?php
      function parametro_valor($parametro)
      {
         $parametro = "Seu valor foi alterado!";
      }
      function parametro_referencia(&$parametro)
      {
         $parametro = "Seu valor foi alterado!";
      }

      if($_POST[dado] != "")              //testa se o usuário digitou o dado
      {
         $aux = $_POST[dado];             //variável auxiliar recebe o valor digitado
         parametro_valor($aux);
         echo("Valor da variável \$aux depois de chamar a função parametro_valor( ): $aux<br>");
         parametro_referencia($aux);
         echo("Valor da variável \$aux depois de chamar a função parametro_referencia( ): $aux<br>");
       }
   ?>
</body>
</html> <!-- nome do arquivo exemplo_funcoes.php -->
```

Funções próprias do PHP

- Assim como em outras linguagens de programação, o PHP possui uma vasta quantidade de funções próprias da linguagem para executar funcionalidades diversas. Em nosso curso veremos apenas algumas dessas funções.
  - ***Funções do PHP que serão vistas no curso:***
    - date();
    - is_numeric();
    - strlen();
    - substr();
    - explode();

Função date()

- A função date(); do PHP retorna data ou hora local(do servidor), depende dos parâmetros que ela recebe na chamada;

- A função date(); recebe como parâmetros valores String pré-determinados e a partir desses parâmetros ela retorna valores correspondentes.
  - **Exemplo:**

```
<?php
    $dt = date("d/m/Y");  //variavel $dt recebe o retorno da função
    echo($dt);  //apresentará na tela a data atual ex: 03/06/2009
?>
```

Alguns parâmetros para a função date()

Caractere de *format*	Descrição	Exemplo de valores retornados
Day	---	---
d	Dia do mês, 2 digitos com preenchimento de zero	*01* até *31*
D	Uma representação textual de um dia, três letras	*Mon* até *Sun*
j	Dia do mês sem preenchimento de zero	*1* até *31*
l ('L' minúsculo)	A representação textual completa do dia da semana	*Sunday* até *Saturday*
Mês	---	---
F	Um representação completa de um mês, como January ou March	*January* até *December*
m	Representação numérica de um mês, com leading zeros	*01* a *12*
M	Uma representação textual curta de um mês, três letras	*Jan* a *Dec*
n	Representação numérica de um mês, sem leading zeros	*1* a *12*
Year	---	---
Y	Uma representação de ano completa, 4 dígitos	Exemplos: *1999* ou *2003*
y	Uma representação do ano com dois dígitos	Exemplos: *99* ou *03*

Lista completa de parâmetros: http://www.php.net/manual/pt_BR/function.date.php

Função is_numeric()

- Essa função recebe como parâmetro uma variável e retorna **TRUE** se ela for um número ou **FALSE** se ela for uma string.

- ***Exemplo:***

```
<?php
    $aux = "Sou uma string!";            //variável recebe um valor String
    if(is_numeric($aux) == true)
    {
        echo("Ela é um número!");
    }
    else
    {
        echo("Ela NÃO é um número!");
    }
        //NESSE CASO SERÁ IMPRESSO "Ela NÃO é um número!"
?>
```

Função strlen()

- Essa função recebe como parâmetro uma variável do tipo String e retorna o número de caracteres dela (tamanho).

- ***Exemplo:***

```
<?php
    $aux = "Lucas";            //variável recebe valor tipo String

    echo(strlen($aux));        //será impresso 5;
?>
```

Função substr()

- Essa função recebe 3 parâmetros e retorna parte da String especificada.
  - O primeiro parâmetro que ela recebe é a String;
  - O segundo parâmetro é o valor referente ao inicio da parte desejada da String;
  - O terceiro parâmetro é referente ao número de caracteres da parte desejada da String;

- ***Exemplo***

```
<?php
    $aux = "Curso PHP!";            //variável recebe valor String
    echo( substr($aux, 0, 4) );            //será impresso Curso
?>
```

Função explode()

Função explode()

- Ela divide uma String em partes, retornando um array, sendo que cada posição desse array contém uma parte da String dividida.

- Ela recebe dois parâmetros:
  - O primeiro é o delimitador responsável pela divisão;
  - O segundo é a String que vai ser dividida;
- **Exemplo:**

```
<?php
    $dt = "03/06/2009";          //variável recebe um valor String
    $vet = explode("/", $dt);
    echo ($vet[0]."<br>");                //imprime "03"
    echo ($vet[1]."<br>");                //imprime "06"
    echo ($vet[2]."<br>");                //imprime "2009"
?>
```

Inclusão de arquivos

- Em PHP é possível usar funções e variáveis contidas em arquivos diferentes, para isso basta incluir o arquivo onde está a função ou variável, no arquivo atual onde elas serão usadas.

- Essa técnica funciona como se fosse a inclusão de bibliotecas de código.

- Para fazer a inclusão de arquivos, o PHP oferece 4 funções especiais:
  - include;
  - include_once;
  - require;
  - require_once;

Inclusão de arquivos - *INCLUDE_ONCE*

- No nosso curso usaremos basicamente somente a função de inclusão ***include_once*** para evitar erros.

- Se você tentar incluir duas ou mais vezes um mesmo arquivo, é gerado um erro no interpretador PHP, pois é como se você tentasse incluir a mesma biblioteca de códigos mais de uma vez.

- O include_once é inteligente com relação a isso e evita esse erro, pois antes dele fazer a inclusão, ele testa se o arquivo que você está tentando incluir já está incluido. Se ele já estiver incluido, o include_once não irá inclui-lo novamente, mas caso não tiver incluido, ele o incluirá.
  - ***Exemplo prático include_once***

Exemplo com INCLUDE_ONCE

```
<?php
//nome do arquivo  "exemplo_include_once.php"
//ESSE ARQUIVO SERÁ INCLUIDO NO ARQUIVO  exemplo_include_once_imprime.php

   function imprime_data()
   {
      $dt = date("d/m/Y");
      return $dt;
   }

?>

<?php
//nome do arquivo  "exemplo_include_once_imprime.php"
include_once 'exemplo_include_once.php';  //é passado o caminho do arquivo a ser
incluido

   echo(imprime_data());       //chamada da função que é declarada o arquivo que foi
incluido

?>
```

- Módulo III -
PHP OO - Programação Orientada à Objetos com PHP

Desenvolvimento web:
PHP Orientado à Objetos

Lucas Francisco da Matta Vegi
www.mr-bin.blogspot.com

Introdução a orientação à objetos

- A orientação à objetos é uma maneira de programar que trata de modelar os processos de programação de uma maneira próxima à realidade: tratando a cada componente de um programa como um **objeto real**, com suas respectivas características e funcionalidades.

- Um software orientado à objetos é composto basicamente de:
  - Classes
    - Métodos (construtor, set, get...)
    - Atributos
  - Objetos

Introdução a orientação à objetos

- ***Exemplo de classe PHP:***
  - Uma ***classe pessoa*** possui os ***atributos*** :
    - nome
    - cpf
    - rg
    - telefone
    - endereço
    - email
  - Uma ***classe pessoa*** possui os ***métodos*** :
    - construtor
    - SETs
    - GETs
    - cadastrar
    - altera
    - exclui

PHP Orientado à Objetos

- A partir da versão 5, a orientação à objetos do PHP ficou praticamente completa, não deixando nada a desejar com relação as linguagens de programação voltadas para desktop.

- Dentre as funcionalidades que ela suporta, podemos citar Sobrescrita de função, Sobrecarga de função, Herança, Polimorfismo e Interfaces.

- Assim como no Java, PHP OO não suporta herança múltipla. A solução que os programadores fazem para compensar isso é a utilização de interfaces.

PHP Orientado à Objetos

- No nosso curso, a parte de orientação à objeto do PHP que será trabalhada, é composta basicamente de:
  - Criação de classe
  - Atributos
  - Métodos construtor
  - Encapsulamento
    - Modificadores de acesso
      - private
      - public
      - protected
  - Métodos Set
  - Métodos Get
  - Criação de objetos
  - Chamada de métodos
  - Herança

Criação de classes

- Exemplo de criação de classe usuário

```
<?php
    class usuario
    {
        //atributos da função
        //métodos da função
        ...
        ...
        ...
    }
?>
```

Atributos

- Exemplo de criação de atributos da classe usuário

```
<?php
    class usuario
    {
        $idUsuario;        //atributo
        $nome;             //atributo
        $login;            //atributo
        $senha;            //atributo

        //métodos da função

    }
?>
```

Métodos construtores

- **Em PHP, os métodos construtores podem ter *dois* possíveis nomes:**
  - mesmo nome da classe
  - __construct

- **Exemplo de criação do método construtor da classe usuário**

```
<?php
    class usuario
    {
        $idUsuario;        //atributo
        $nome;             //atributo
        $login;            //atributo
        $senha;            //atributo

        function __construct()    //criação do método construtor
        {

        }
        //métodos da função...
    }
?>
```

Encapsulamento

- É um processo no qual se ocultam as características internas de um objeto àqueles elementos que não têm porque conhece-las. Os ***modificadores de acesso*** servem para indicar as permissões que terão outros objetos para acessar a seus métodos e atributos.

- ***Modificadores de acesso em PHP:***
  - public
  - private
  - protected

Modificadores de Acesso PHP

- **public :**
  - É o modificador de acesso padrão, ou seja, quando não se define o modificador de acesso de um método ou atributo, esse é interpretado pelo PHP como public. É o modificador *mais permissivo* , pois os métodos ou atributos public podem ser acessados em qualquer parte do programa.
- **private :**
  - É o modificador de acesso *mais restritivo* . Um atributo ou método private, só pode ser acessado pelo próprio objeto.
- **protected :**
  - É um modificador de *restrição média* . Um atributo ou método protected, pode ser acessado pelo próprio objeto e por objetos das classes filhas (que o herdou).

Modificadores de Acesso PHP - Exemplo

- Exemplo de encapsulamento (uso de modificadores de acesso)

```
<?php
    class usuario
    {
        private $idUsuario;        //atributo privado
        private $nome;             //atributo privado
        private $login;            //atributo privado
        private $senha;            //atributo privado

        public function __construct()    //criação do método construtor público
        {

        }
        //métodos da função...
    }
?>
```

Métodos SET

- Cada ***atributo private*** de uma classe, deve possuir um ***método public SET*** correspondente para que possam ser atribuidos valores a ele através desse método.

```
<?php
    class usuario
    {
        private $idUsuario;        //atributo privado
        private $nome;             //atributo privado
        private $login;            //atributo privado
        private $senha;            //atributo privado

        public function __construct()      //criação do método construtor público
        {

        }

        public function setNome($n)
        {
            $this->nome = $n;      //atributo $nome recebe o valor referente a $n
        }
    }
?>
```

Métodos GET

- Cada ***atributo private*** de uma classe, deve possuir um ***método public GET*** correspondente para que possam ser recuperados seus valores através desse método.

```
<?php
    class usuario
    {
        private $idUsuario;        //atributo privado
        private $nome;             //atributo privado
        private $login;            //atributo privado
        private $senha;            //atributo privado

        public function __construct()      //criação do método construtor público
        {

        }

        public function getNome()
        {
            return $this->nome;      //retorna valor do atributo $nome
        }
    }
?>
```

Criação de objetos

Exemplo de criação de objeto da ***classe usuario:***

```
<?php
   //nome do arquivo cria_objeto.php

   include_once "usuario.class.php";  //inclui classe usuario

   $objeto = new usuario();   //cria objeto da classe pessoa
?>
```

Chamada de métodos

Exemplo de chamada de métodos da ***classe usuario:***

```
<?php
   //nome do arquivo cria_objeto.php

   include_once "usuario.class.php";  //inclui classe usuario

   $objeto = new usuario();   //cria objeto da classe pessoa
   $aux = "Lucas Vegi";          //variável auxiliar recebe nome
   $objeto->setNome($aux);     //chama método setNome
   echo( $objeto->getNome() );     //chama método getNome
?>
```

Herança em PHP

- Com a Herança, as classes podem herdar as características (atributos e métodos) de outras, de modo que se podem fazer ***classes especializadas*** , baseadas em outras mais gerais.

- ***Exemplo de uma classe herdando outra:***

```
<?php
  include_once "usuario.class.php";   //código da classe pai deve ser incluido na filha
  class admin extends usuario         //classe admin herda da classe usuario
  {
      private $setor;
      public function __construct()
      {
        parent::__construct();        //chama construtor da classe pai
      }
  }
?>
```

Herança em PHP

- Exemplo de chamada de métodos da ***classe admin*** que foram **herdados da classe usuario*:***

```
<?php
  //nome do arquivo cria_objeto_heranca.php

  include_once "admin.class.php";  //inclui classe admin

  $objeto = new admin();   //cria objeto da classe admin
  $aux = "Lucas Vegi";          //variável auxiliar recebe nome
  $objeto->setNome($aux);     //chama método setNome
  echo( $objeto->getNome() );     //chama método getNome
?>
```

Exemplo de uma aplicação em PHP OO

- A partir de agora começaremos a desenvolver uma aplicação chamada Blog' IS.
- Essa aplicação basicamente tem a funcionalidade de um blog onde um usuário poderá postar mensagens.
- Até o término desse curso, essa aplicação será aperfeiçoada, ganhando cada vez mais funcionalidades.

Mãos a obra!

```php
<?php
class post    //arquivo post.class.php
{
    private $titulo;
    private $texto;
    private $momento;

    public function __construct()
    {

    }

    public function getTitulo()
    {
        return $this->titulo;
    }

    public function getTexto()
    {
        return $this->texto;
    }
```

```php
    //continuação da classe.....
    public function getMomento()
    {
        return $this->momento;
    }

    public function setTitulo($tit)
    {
        $this->titulo = $tit;
    }

    public function setTexto($tex)
    {
        $this->texto = $tex;
    }

    public function setMomento()
    {
        $this->momento = date("H:i:s");
    }
}
?>
```

```
<?php
                                        //arquivo home.php
include_once 'post.class.php';
session_start();
echo("
<html>
<head><title>Blog'IS - Information System</title></head>
<body>");
echo("
<table align='center' border='0' width='780' cellspacing='5'>
   <tr>
      <td align='center' colspan='2'><a href='home.php'><img src='blogis.jpg'
border='0'/></a></td>
   </tr>
   <tr>
      <td align='center' bgcolor='#86a7f6'><font face='verdana' size='3' color='#FFFFFF'><b>Seja
bem vindo</b></font></td>
     </tr>
      <form action='' method='POST'>
   <tr>
      <td align='center'><input type='text' name='titulo'></td>
   </tr>
   <tr>
      <td align='center'><textarea name='texto'></textarea></td>
   </tr>
```

```
<tr>
      <td align='center'><input type='submit' value='POSTAR!'></td>
   </tr> </form>
");
   if(isset($_POST['titulo']) && isset($_POST['texto']))
   {
      $postagem = new post();
      $postagem->setTitulo($_POST['titulo']);
      $postagem->setTexto($_POST['texto']);
      $postagem->setMomento();
      echo("<tr>
            <td align='center' bgcolor='#86a7f6'><font face='verdana' size='3'
color='#FFFFFF'><b>Postagens</b></font></td>
         </tr>");
      echo("<table border=0 align='center' cellspacing='0' cellpadding='0'>");
      echo("    <tr><td align='center'><font size='5' face='arial'><b><u>".$postagem->getTitulo()."
</u></b></font><br><br></td></tr>
            <tr><td align='center'><font size='3' face='arial'>".$postagem->getTexto()."
</font></td></tr>
            <tr><td align='center'><font size='2' face='arial'>horário - ".$postagem-
>getMomento()."</font><br><hr><br><br></td></tr>
             </table>
           ");
   }
echo("</table></body></html>");
?>
```

- Módulo IV -

Solução para os dados voláteis: Sessão

Desenvolvimento web:

PHP Orientado à Objetos

Lucas Francisco da Matta Vegi
www.mr-bin.blogspot.com

Dados são mais voláteis em PHP

- Em PHP, assim que um script é executado pelo servidor e retornado para o cliente em forma de HTML, ***todos os valores contidos nas variáveis e objetos são perdidos*** , pois eles são retirados da memória RAM dos servidores.

- Isso se deve ao fato de que um servidor web executa ao mesmo tempo milhares de scripts oriundos de toda parte do mundo, portanto eles devem ***otimizar o uso de memória RAM,*** evitando o desperdício.

- ***Tudo bem, mas e se eu quiser manter meus dados na memória ram dos servidores por mais tempo???***

Uso de SESSÃO

- Sessões são ***vetores associativos*** especiais do PHP que reservam um espaço na memória RAM dos servidores e assim **tornam os dados menos voláteis.**

- Os dados mantidos em uma sessão permanecem na memória RAM dos servidores enquanto o navegador do cliente(browser) estiver aberto ou enquanto a sessão não expirar por inatividade ou por vencer sua vida útil.

Iniciando SESSÃO

- Pra começar a trabalhar com uma sessão, você precisa primeiro aprender uma coisa. ***Iniciar a sessão*** . Mesmo que você ainda não saiba o que vai colocar nela. Vamos por partes.

```
<?php
    session_start();        //inicia a sessão
?>
```

Tempo de vida de uma SESSÃO

- Um sessão, por padrão tem o ***TTL*** (time to live ou tempo de vida) de ***180 minutos*** . Se você precisa por algum motivo que sua sessão expire em menor ou maior tempo, use a função **session_cache_expire(** minutos **);**

```
<?php
    session_start();                    //inicia a sessão
    session_cache_expire(10);  //sessão dura 10 minutos
?>
```

Registrando uma SESSÃO

- Digamos que você quer manter algum dado na sessão, para posterior uso.

  Vamos para um exemplo prático.

```
<?php
    session_start();                    //inicia a sessão
    $nome = "Lucas Vegi";               //variável recebe valor string

    $_SESSION['nome'] = $nome;   //registra sessão

    echo("Valor da sessão: ".$_SESSION['nome']);    //imprime valor
?>
```

Verificando se uma SESSÃO existe

- O PHP oferece um método especial para verificar se uma sessão foi registrada, esse método chama-se **isset()** e retorna um valor **boolean**.

```
<?php
    session_start();                    //inicia a sessão

    if( isset($_SESSION['nome']) == true) //testa se sessão  existe
    {
      echo("Sessão existe!");
   }
   else{   echo("Sessão não existe!");     }
?>
```

Destruindo uma SESSÃO

- Agora que você sabe como registrar, expirar, verificar se existe, vamos ***destuir*** uma sessão. Para isso usaremos o método especial do PHP chamado **unset()**

```
<?php

    session_start();                    //inicia a sessão

    unset($_SESSION['nome']);           //destroi a sessão

?>
```

Implementação do Blog'IS com sessão

- A mesma aplicação desenvolvida anteriormente, agora será aperfeiçoada consideravelmente com a utilização de sessão e dessa forma ficará muito mais fácil entender esses conceitos.

- O sistema será composto das classes:
  - post
  - usuario

- O sistema se comportará da seguinte forma:
  - **Caso o usuário estiver logado:**
    - Ele deve mostrar o nome do usuário logado,
    - Mostra a opção de postar conteúdo
    - Mostra postagens já realizadas
  - **Caso contrário:**
    - Mostra a opção "cadastrar".

```php
<?php
class post
{
   private $titulo;
   private $texto;
   private $momento;

   public function __construct()
   {

   }

   public function getTitulo()
   {
      return $this->titulo;
   }

   public function getTexto()
   {
      return $this->texto;
   }

   public function getMomento()
   {
      return $this->momento;
   }

   public function setTitulo($tit)
   {
     $this->titulo = $tit;
   }

   public function setTexto($tex)
   {
      $this->texto = $tex;
   }

   public function setMomento()
   {
      $this->momento = date("H:i:s");
   }

   public function publicar()
   {
   $_SESSION['posts'][$_SESSION['contador_posts']]['titulo'] = $this->titulo;
   $_SESSION['posts'][$_SESSION['contador_posts']]['texto'] = $this->texto;
   $_SESSION['posts'][$_SESSION['contador_posts']]['momento'] = $this->momento;
   $_SESSION['contador_posts']+=1;
   }
}

//arquivo post.class.php

?>
```

```php
<?php
class usuario
{
    private $nome;
    private $email;
    public function __construct()
    {

    }

    public function getNome()
    {
        return $this->nome;
    }

    public function setNome($n)
    {
        $this->nome = $n;
    }

    public function getEmail()
    {
        return $this->email;
    }

    public function setEmail($n)
    {
        $this->email = $n;      //atributo $email recebe o valor referente a $n
    }

    public function cadastrar()
    {
        session_start();
        $_SESSION['usuario']= $this->nome;
        $_SESSION['email']= $this->email;
        $_SESSION['contador_posts']=0;

        echo "<tr>
                <td align='center' bgcolor='#86a7f6' width='50%'><a href='me-cadastrar.php' style='text-decoration: none;'><font face='verdana' size='3' color='#FFFFFF'><b>Me cadastrar!</b></font></a></td>
                <td align='center' bgcolor='#86a7f6' width='50%'><a href='me-logar.php' style='text-decoration: none;'><font face='verdana' size='3' color='#FFFFFF'><b>Logar</b></font></a></td>
            </tr>
            <tr>
                <td align='center' colspan='2'><br><br><br><font face='verdana' size='3' color='#7097f2'><b>Usuário cadastrado com sucesso!</b></font></td>
            </tr>
            <tr>
                <td align='center' colspan='2'><a href='home.php'><font face='verdana' size='3' color='#0344e1'><b>Faça já os seus POSTs!</b></font></a></td>
            </tr>
            ";
    }
}       //arquivo usuario.class.php
?>
```

```php
<?php
//nome do arquivo me-cadastrar.php
include_once "usuario.class.php";
session_start();
echo("
<html>
<head><title>Blog'IS - Information System</title></head>
<body>");
echo("
<table align='center' border='0' width='780' cellspacing='5'>
    <tr>
        <td align='center' colspan='2'><a href='home.php'><img src='blogis.jpg' border='0'/></a></td>
    </tr>
");

if($_SESSION['usuario'])
{
    session_destroy();
    session_start();
}

if(($_POST['nome']) && ($_POST['email']))
{
    $user = new usuario();
    $user->setNome($_POST['nome']);
    $user->setEmail($_POST['email']);
    $user->cadastrar();
}
```

```
else
{
   echo("
         <tr>
            <td align='center' bgcolor='#86a7f6' width='50%'><a href='me-cadastrar.php' style='text-decoration:
none;'><font face='verdana' size='3' color='#FFFFFF'><b>Me cadastrar!</b></font></a></td>
            <td align='center' bgcolor='#86a7f6' width='50%'><a href='#' style='text-decoration: none;'><font
face='verdana' size='3' color='#FFFFFF'><b>Logar</b></font></a></td>
         </tr>
         <tr>
            <form action='' method='POST'>
            <td align='center' colspan='2'><br><br><br><font face='arial' size='2'><b>Nome:</b></font> 
 <input type='text' size='20' name='nome'></td>
         </tr>
         <tr>
            <td align='center' colspan='2'><font face='arial' size='2'><b>Email:</b></font>  <input
type='text' size='20' name='email'></td>
         </tr>
         <tr>
            <td align='center' colspan='2'><input type='submit' value='Cadastre-me'></td>
         </tr>
         </form>
      ");
}
echo("</table></body></html>");
//fim do arquivo me-cadastrar.php
?>
```

```php
<?php
                                        //nome do arquivo home.php
include_once 'post.class.php';
session_start();
echo("
<html>
<head><title>Blog'IS - Information System</title></head>
<body>");
echo("
<table align='center' border='0' width='780' cellspacing='5'>
   <tr>
      <td align='center' colspan='2'><a href='home.php'><img src='blogis.jpg' border='0'/></a></td>
   </tr>
");

if($_GET['opcao']=='sair')
{
   unset($_SESSION);
   session_destroy();
}
```

```
if(isset($_SESSION['usuario']))
{
   echo "<tr>
         <td align='center' bgcolor='#86a7f6'><font face='verdana' size='3' color='#FFFFFF'><b>Usuário logado:
</b><i>".$_SESSION['usuario']."</i> - <a href='?opcao=sair'>Sair</a></font></td>
      </tr>
      <form action='' method='POST'>
      <tr>
         <td align='center'><input type='text' name='titulo'></td>
      </tr>
      <tr>
         <td align='center'><textarea name='texto'></textarea></td>
      </tr>
      <tr>
         <td align='center'><input type='submit' value='POSTAR!'></td>
      </tr>
      </form>
      ";

   if(isset($_POST['titulo']) && isset($_POST['texto']))
   {
      $postagem = new post();
      $postagem->setTitulo($_POST['titulo']);
      $postagem->setTexto($_POST['texto']);
      $postagem->setMomento();
      $postagem->publicar();
   }
```

```
   if(isset($_SESSION['posts']))
   {
      echo("<tr>
            <td align='center' bgcolor='#86a7f6'><font face='verdana' size='3'
color='#FFFFFF'><b>Postagens</b></font></td>
          </tr>");
      echo("<table border=0 align='center' cellspacing='0' cellpadding='0'>");
      for($i=0;$i<count($_SESSION['posts']);$i++)
      {
         echo "<tr><td align='center'><font size='5' face='arial'><b><u>".$_SESSION['posts'][$i]['titulo']."
</u></b></font><br><br></td></tr>
              <tr><td align='center'><font size='3' face='arial'>".$_SESSION['posts'][$i]['texto']."</font></td></tr>
              <tr><td align='center'><font size='2' face='arial'>publicado por ".$_SESSION['usuario']." - ".$_SESSION
['posts'][$i]['momento']."</font><br><hr><br><br></td></tr>";
      }
      echo("</table>");
   }
}
else
{
   echo "<tr>
         <td align='center' bgcolor='#86a7f6' width='50%'><a href='me-cadastrar.php' style='text-decoration:
none;'><font face='verdana' size='3' color='#FFFFFF'><b>Me cadastrar!</b></font></a></td>
         <td align='center' bgcolor='#86a7f6' width='50%'><a href='#' style='text-decoration: none;'><font
face='verdana' size='3' color='#FFFFFF'><b>Logar</b></font></a></td>
       </tr>";
}
echo("</table></body></html>");            //fim do arquivo home.php
?>
```

Mesmo com o uso de SESSÃO os dados continuam voláteis

- Vocês devem ter observado que mesmo com o uso de sessão, os dados continuam voláteis após o fechamento do navegador.

- Isso se deve ao fato de que a Sessão simula no servidor, uma área memória equivalente à memória RAM desktop, que é volátil. Para resolver essa limitação e guardar os dados em definitivo, somente com o uso de um Banco de Dados.

- Além de guardar os dados em definitivo, o uso do banco de dados proporcionaria que os usuários tivessem acesso a todas as postagens de todos os usuários e não somente às suas.

Dê sua opinião sobre o curso!

- Entrem por favor no link abaixo e respondam algumas poucas perguntas com relação ao curso. Sua contribuição será muito importante!

http://spreadsheets.google.com/viewform?formkey=ckx2WlVhaU5Pd2dwX2FOVjFiNEhhYlE6MA..

Contatos

lucasvegi@gmail.com

Referências Bibliográficas

- ALVES, L. L; BITTENCOUT, F. R. PHP: Conceitos Essenciais para Implementação de Aplicações Web. 7 Faces, Itabira, MG, v. 4, p. 193-208, 2003.
- http://apostilas.fok.com.br/manual-do-php/reserved.php
- http://www.php.net
- http://www.revistaphp.com.br/artigo.php?id=79

Apostila de PHP

Apostila desenvolvida por Bruno Rodrigues Siqueira(bruno@netfly.com.br)

Índice

ÍNDICE.....1

1.INTRODUÇÃO.....10

O QUE É PHP?.....10
HISTÓRIA DO PHP.....11
O QUE PODE SER FEITO COM PHP?.....12
TABELA COMPARATIVA.....12

2. SINTAXE BÁSICA.....13

DELIMITANDO O CÓDIGO PHP.....13
SEPARADOR DE INSTRUÇÕES.....13
NOMES DE VARIÁVEIS.....14
COMENTÁRIOS.....14

3. TIPOS.....16

TIPOS SUPORTADOS.....16
Inteiros (integer ou long).....16
Números em Ponto Flutuante (double ou float).....16
Strings.....17
Arrays.....18
LISTAS.....18
OBJETOS.....19
BOOLEANOS.....19
TRANSFORMAÇÃO DE TIPOS.....19
COERÇÕES.....20
TRANSFORMAÇÃO EXPLÍCITA DE TIPOS.....20
COM A FUNÇÃO SETTYPE.....21

4. CONSTANTES.....22

CONSTANTES PRÉ-DEFINIDAS.....22
DEFININDO CONSTANTES.....22

5. OPERADORES.....23

ARITMÉTICOS.....23
DE STRINGS.....23
DE ATRIBUIÇÃO.....23
BIT A BIT.....24
LÓGICOS.....24
COMPARAÇÃO.....24

Expressão condicional....25
de incremento e decremento....25
Ordem de precedência dos operadores....25

6. ESTRUTURAS DE CONTROLE....27

Blocos....27
Comandos de seleção....27
if....27
switch....30
comandos de repetição....31
while....31
do... while....32
for....33
Quebra de fluxo....34
Break....34
Continue....34

7. FUNÇÕES....36

Definindo funções....36
Valor de retorno....36
Argumentos....36
Passagem de parâmetros por referência....37
Argumentos com valores pré-definidos (default)....38
Escopo....39

8. VARIÁVEIS....41

O modificador static....41
Variáveis Variáveis....42
Variáveis enviadas pelo navegador....42
URLencode....42
Variáveis de ambiente....43
Verificando o tipo de uma variável....43
Função que retorna o tipo da variável....43
Funções que testam o tipo da variável....43
Destruindo uma variável....44
Verificando se uma variável possui um valor....44
Arrays Multidimensionais....45

9. CLASSES E OBJETOS....46

Classe....46
Objeto....46
A variável $this....46
SubClasses....47
Construtores....47

10. PHP AVANÇADO....49

Interagindo com o browser....49
Utilizando formulários HTML....49
Cookies....51

SESSÃO....51
REQUIRE....52
INCLUDE....52
FORMULÁRIOS AVANÇADOS....52
ARQUIVOS REMOTOS....53
Lendo....53
Escrevendo....53
TRATAMENTO DE ERROS....54

11.SQL....55

O QUE É?....55
SELECT....55
Subseleções....56
INSERT....56
UPDATE....56
DELETE....57
O QUE EM COMUM ENTRE: DELETE X UPDATE....57
CREATE....57
DROP....58
ALTER....58

12. BANCOS DE DADOS COMPATÍVEIS COM O PHP:....58

13. BIBLIOTECA DE FUNÇÕES....59

BIBLIOTECAS REQUERIDAS....59
ARRAY....59
Array....59
array_keys [PHP4]....59
Array_merge [PHP4]....60
Array_pop[PHP4]....60
Array_push[PHP4]....60
Array_shift[PHP4]....60
Array_slice[PHP4]....60
Array_splice[PHP4]....60
Array_unshift[PHP4]....61
Array_values[PHP4]....61
Array_walk....61
Arsort....61
Asort....61
Compact[PHP4]....61
Count....62
Current....62
Each....62
End....62
Extract....62
In_array[PHP4]....63
Key....63
Ksort....63
List....63
Next....63
Pos....63
Prev....63

Range....63
Reset....64
Rsort....64
Shuffle....64
Sizeof....64
Sort....64
Uasort....64
Uksort....64
Usort....65
MATEMÁTICA PARA NÚMEROS INTEIROS....65
Bcpow....65
Bcscale....65
DATAS....65
Checkdate....65
Date....65
Getdate....66
Gettimeofday....66
Gmdate....67
Gmmktime....67
Gmstrftime....67
Microtime....67
Mktime....67
Strftime....67
Time....68
DIRETÓRIO....68
Chdir....68
Classe dir....69
Closedir....69
Opendir....69
Readdir....69
Rewinddir....69
EXECUÇÃO DE PROGRAMAS....69
Escapeshellcmd....69
Exec....69
Passthru....70
System....70
SISTEMA DE ARQUIVOS DO SERVIDOR....70
Basename....70
Chgrp....70
Chmod....70
Chown....71
Clearstatcache....71
Copy....71
Delete....71
Dirname....71
Diskfreespace....71
Fclose....71
Feof....71
Fgetc....72
Fgetcsv....72
Fgets....72
Fgetss....72
File....72
File_exists....72
Fileatime....72

Filectime....73
Filegroup....73
Fileinode....73
Filemtime....73
Fileowner....73
Fileperms....73
Filesize....73
Filetype....74
Flock....74
Fopen....74
Fpassthru....75
Fputs....75
Fread....75
Fseek....75
Ftell....75
Fwrite....75
Is_dir....76
Is_executable....76
Is_file....76
Is_link....76
Is_readable....76
Is writeable....76
Link....76
Linkinfo....77
Mkdir....77
Pclose....77
Popen....77
Readfile....77
Readlink....77
Rename....77
Rewind....77
Rmdir....78
Set_file_buffer....78
Stat....78
Symlink....78
Tempnam....78
Touch....78
Umask....79
Unlink....79
OPÇÕES E INFORMAÇÕES DO PHP....79
Error_log....79
Error_reporting....79
Extension_loaded....80
Get_cfg_var....80
Get_current_user....80
Get_magic_quotes_gpc....80
Get_magic_quotes_runtime....80
Getenv....80
Getlastmod....80
Getmyinode....81
Getmypid....81
Getmyuid....81
Getrusage....81
Phpinfo....81
Phpversion....81
Putenv....81

Set_magic_quotes_runtime....81
Set_time_limit....81
MATEMÁTICA....82
Abs....82
Acos....82
Asin....82
Atan....82
Atan2....82
Base_convert....82
Bindec....82
Ceil....82
Cos....83
Decbin....83
Dechex....83
Decoct....83
Exp....83
Floor....83
Getrandmax....83
Hexdec....83
Log....83
Log10....83
Max....84
Min....84
Mt_rand....84
Mt_srand....84
Mt_getrandmax....84
Number_format....84
Octdec....84
Pi....84
Pow....84
Rand....85
Round....85
Sin....85
Sqrt....85
Srand....85
Tan....85
CRIPTOGRAFIA....85
Mcrypt_cbc....85
Mcrypt_cfb....85
Mcrypt_create_iv....86
Mcrypt_ecb....86
Mcrypt_get_cipher_name....86
Mcrypt_get_block_size....86
Mcrypt_get_key_size....86
Mcrypt_ofb....86
FUNÇÕES DIVERSAS....86
Connection_aborted....86
Connection_status....86
Connection_timeout....86
Dl....87
Eval....87
Die....87
Exit....87
Function_exists....87
Ignore_user_abort....87

Iptcparse....87
Leak....87
Mail....88
Pack....88
Register_shutdown_function....88
Serialize....88
Sleep....88
Unpack....88
Unserialize....89
Uniquid....89
Usleep....89
REDE....89
Checkdnsrr....89
Closelog....89
Debugger_on....89
Debugger_off....89
Fsockopen....89
Gethostbyaddr....89
Gethostbyname....90
Openlog....90
Pfsockopen....90
Set_socket_blocking....90
Syslog....90
EXPRESSÕES REGULARES....90
Ereg....90
Ereg_replace....90
Eregi....91
Eregi_replace....91
Split....91
Sql_regcase....91
TRATAMENTO DE SESSÕES....91
Session_decode[PHP4]....91
Session_destroy[PHP4]....91
Session_encode[PHP4]....91
Session_start[PHP4]....91
Session_id[PHP4]....92
Session_is_registered[PHP4]....92
Session_module_name[PHP4]....92
Session_name[PHP4]....92
Session_register[PHP4]....92
Session_save_path[PHP4]....92
Session_unregister[PHP4]....92
STRINGS....92
Addslashes....92
Bin2hex....93
Chop....93
Chr....93
Chunk_split....93
Convert_cyr_string....93
Crypt....93
Echo....93
Explode....93
Flush....94
Get_meta_tags....94
Htmlentities....94

Htmlspecialchars....94
Implode....94
Join....94
Ltrim....94
Md5....94
Nl2br....95
Ord....95
Parse_str....95
Print....95
Printf....95
Quoted_printable_decode....95
Quotemeta....95
Rawurldeconde....95
Rawurlenconde....96
Setlocale....96
Similar_text....96
Soundex....96
Sprintf....96
Strchr....96
Strcmp....96
Strcspn....96
Strip_tags....97
Stripslashes....97
Strlen....97
Strpos....97
Strrpos....97
Strrchr....97
Strrev....97
Strspn....97
Strstr....97
Strtok....98
Strtolower....98
Strtoupper....98
Str_replace....98
Strtr....98
Substr....98
Trim....98
Ucfirst....99
Ucwords....99
FUNÇÕES PARA VARIÁVEIS....99
Doubleval....99
Empty....99
Gettype....99
Intval....99
Is_array....99
Is_double....99
Is_float....99
Is_int....100
Is_integer....100
Is_long....100
Is_object....100
Is_real....100
Is_string....100
Isset....100
Settype....100
Strval....101

Unset.....101
14. Referências na Internet.....101

1. Introdução

O que é PHP?

PHP significa: *Hypertext Preprocessor*. Realmente, o produto foi originalmente chamado de "Personal Home Page Tools"; mas como se expandiu em escopo, um nome novo e mais apropriado foi escolhido por votação da comunidade. Você pode utilizar qualquer extensão que desejar para designar um arquivo PHP, mas os recomendados foram .php , .phtml. O PHP está atualmente na versão 4, chamado de PHP4 ou, simplesmente de PHP.

PHP é uma linguagem de criação de scripts embutida em HTML no servidor. Os produtos patenteados nesse nicho do mercado são as Active Server Pages da Microsoft, o Coldfusion da Allaire e as Java Server Pages da Sun. PHP é às vezes chamado de "o ASP de código-fonte aberto" porque sua funcionabilidade é tão semelhante ao produto/conceito, ou o que quer que seja, da Microsoft.

Exploraremos a criação de script no servidor, mais profundamente, nos próximos capítulos, mas, no momento, você pode pensar no PHP como uma coleção de supertags de HTML que permitem adicionar funções do servidor às suas páginas da Web. Por exemplo, você pode utilizar PHP para montar instantaneamente uma complexa página da Web ou desencadear um programa que automaticamente execute o débito no cartão de crédito quando um cliente realizar uma compra.

Falando estritamente, o PHP tem pouca relação com layout, eventos ou qualquer coisa relacionada à aparência de uma página da Web. De fato, a maior parte do que o PHP realiza é invisível para o usuário final. Alguém visualizando uma página de PHP não será capaz de dizer que não foi escrita em HTML, porque o resultado final do PHP é HTML.

O PHP é um módulo oficial do servidor http Apache, o líder do mercado de servidores Web livres que constitui aproximadamente 55 por cento da World Wide Web. Isso significa que o mecanismo de script do PHP pode ser construído no próprio servidor

Web, tornando a manipulação de dados mais rápida. Assim como o servidor Apache, o PHP é compatível com várias plataformas, o que significa que ele executa em seu formato original em várias versões do UNIX e do Windows. Todos os projetos sob a égide da Apache Software Foundation – incluindo o PHP – são software de código-fonte aberto.

As várias versões do PHP foram aclamadas e premiadas nos últimos anos. O PHP3 foi o finalista em 1999 no LinuxWorld Editor´s Choice Awards (na categoria de biblioteca/ ferramentas de programação) e ganhou o segundo lugar, perdendo só para o ColdFusion, em 1998 no Cnet Builder.com Product Awards (na categoria de melhor ferramenta de script de servidor – eles deram bastante importância ao IDE), ao passo que a combinação PHP3/ MySQL ganhou prêmio de banco de dados do ano no Web98. Nada mau para um software sem relações públicas, sem publicidade e sem uma significativa exposição na mídia.

História do PHP

Rasmus Lerdorf – engenheiro de software, membro da equipe Apache e o homem misterioso do ano – é o criador e a força motriz original por trás do PHP. A primeira parte do PHP foi desenvolvida para utilização pessoal no final de 1994. Tratava-se de um *wrapper* de PerlCGI que o auxiliava a monitorar as pessoas que acessavam o seu site pessoal. No ano seguinte, ele montou um pacote chamado de Personal Home Page Tools (também conhecido como PHP Construction Kit) em resposta à demanda de usuários que por acaso ou por relatos falados depararam-se com o seu trabalho. A versão 2 foi logo lançada sob o título de PHP/FI e incluía o Form Interpreter, uma ferramenta para analisar sintaticamente consultas de SQL.

Em meados de 1997, o PHP estava sendo utilizado mundialmente em aproximadamente 50.000 sites. Obviamente estava se tornando muito grande para uma única pessoa administrar, mesmo para alguém concentrado e cheio de energia como Rasmus. Agora uma pequena equipe central de desenvolvimento mantinha o projeto sobre o modelo de "junta benevolente" do código-fonte aberto, com contribuições de desenvolvedores e usuários em todo o mundo. Zeev Suraski e Andi Gutmans, dois programadores israelenses que desenvolveram os analisadores de sintaxe PHP3 e PHP4, também generalizaram e estenderam seus trabalhos sob a rubrica de Zend.com (Zeev, Andi, Zend, entendeu?).

O quarto trimestre de 1998 iniciou um período de crescimento explosivo para o PHP, quando todas as tecnologias de código-fonte aberto ganharam uma publicidade intensa. Em outubro de 1998, de acordo com a melhor suposição, mais de 100.000 domínios únicos utilizavam PHP de alguma maneira. Um ano depois, o PHP quebrou a marca de um milhão de domínios. Enquanto escrevo esta apostila, o número explodiu para cerca de dois milhões de domínios.

O que pode ser feito com PHP?

Basicamente, qualquer coisa que pode ser feita por algum programa CGI pode ser feita também com PHP, como coletar dados de um formulário, gerar páginas dinamicamente ou enviar e receber *cookies*.

PHP também tem como uma das características mais importantes o suporte a um grande número de bancos de dados, como dBase, Interbase, mSQL, mySQL, Oracle, Sybase, PostgreSQL e vários outros. Construir uma página baseada em um banco de dados torna-se uma tarefa extremamente simples com PHP.

Além disso, PHP tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente, HTTP. Ainda é possível abrir *sockets* e interagir com outros protocolos.

Tabela comparativa

Custos Comparativos				
Item	**ASP**	**Cold Fusion**	**JSP**	**PHP**
Desenvolvimento	US$ 0 – 480	US$ 395	US$ 0	US$ 0
Servidor	US$ 620	US$ 1.295	US$ 0 – 595	US$ 0
RDBMS	US$ 1.220 - 4220	US$ 0 - ~10.000	US$ 0 - ~10.000	US$ 0
Suporte de incidente	US$0 - 245	US$ 0 - 75	US$ 0 - 75	US$ 0

2. Sintaxe Básica

Delimitando o código PHP

O código PHP fica embutido no próprio HTML. O interpretador identifica quando um código é PHP pelas seguintes tags:

```
<?php
comandos
?>

<script language="php">
comandos
</script>

<?
comandos
?>

<%
comandos
%>
```

O tipo de *tags* mais utilizado é o terceiro, que consiste em uma "abreviação" do primeiro. Para utilizá-lo, é necessário habilitar a opção *short-tags* na configuração do PHP. O último tipo serve para facilitar o uso por programadores acostumados à sintaxe de ASP. Para utilizá-lo também é necessário habilitá-lo no arquivo de configuração do PHP (*php.ini*)

Separador de instruções

Para cada fim de linha de código tem que haver um ponto e vírgula, indicando ao sistema fim de instrução.

Exemplo.

```
<?
echo 'com ponto e vírgula' ;
?>
```

Linhas de comando, de controle, não precisam de ponto e vírgula.

Exemplo.:

```
<?
if ($x == $x){  //aqui não precisa de ponto e vírgula
echo 'com ponto e vírgula' ; //aqui precisa de ponto e vírgula
}
?>
```

Nomes de variáveis

Toda variável em PHP tem seu nome composto pelo caracter $ e uma string, que deve iniciar por uma letra ou o caracter "_". **PHP é case sensitive**, ou seja, as variáveis $php e $PHP são diferentes. Por isso é preciso ter muito cuidado ao definir os nomes das variáveis. É bom evitar os nomes em maiúsculas, pois como veremos mais adiante, o PHP já possui alguma variáveis pré-definidas cujos nomes são formados por letras maiúsculas.

Comentários

Há dois tipos de comentários em código PHP:

Comentários de uma linha:

Marca como comentário até o final da linha ou até o final do bloco de código PHP - o que vier antes. Pode ser delimitado pelo caracter "#" ou por duas barras (//).

```
Exemplo:
<?  echo "teste"; #isto é um teste
echo "teste"; //este teste é similar ao anterior
?>
```

Comentários de mais de uma linha:

Tem como delimitadores os caracteres "/*" para o início do bloco e "*/" para o final do comentário. Se o delimitador de final de código PHP (?>) estiver dentro de um comentário, não será reconhecido pelo interpretador.

Exemplos:

```
<?
echo “teste”; /* Isto é um comentário com mais
de uma linha que funciona corretamente
*/
?>
```

3. Tipos

Tipos Suportados

PHP suporta os seguintes tipos de dados:

- Inteiro
- Ponto flutuante
- String
- Array
- Objeto

PHP utiliza checagem de tipos dinâmica, ou seja, uma variável pode conter valores de diferentes tipos em diferentes momentos da execução do script. Por este motivo não é necessário declarar o tipo de uma variável para usá-la. O interpretador PHP decidirá qual o tipo daquela variável, verificando o conteúdo em tempo de execução.

Ainda assim, é permitido converter os valores de um tipo para outro desejado, utilizando o *typecasting* ou a função `settype` (ver adiante).

Inteiros (integer ou long)

Uma variável pode conter um valor inteiro com atribuições que sigam as seguintes sintaxes:

```
$php = 1234;    # inteiro positivo na base decimal
$php = -234;    # inteiro negativo na base decimal
$php = 0234;    # inteiro na base octal-simbolizado pelo 0
                # equivale a 156 decimal
$php = 0x34;    # inteiro na base hexadecimal(simbolizado
                # pelo 0x) - equivale a 52 decimal.
```

A diferença entre inteiros simples e `long` está no número de bytes utilizados para armazenar a variável. Como a escolha é feita pelo interpretador PHP de maneira transparente para o usuário, podemos afirmar que os tipos são iguais.

Números em Ponto Flutuante (double ou float)

Uma variável pode ter um valor em ponto flutuante com atribuições que sigam as seguintes sintaxes:

```
$php = 1.234;
$php = 23e4;    # equivale a 230.000
```

Strings

Strings podem ser atribuídas de duas maneiras:

a) utilizando aspas simples (') - Desta maneira, o valor da variável será exatamente o texto contido entre as aspas (com exceção de \\ e \' - ver tabela abaixo)
b) utilizando aspas duplas (") - Desta maneira, qualquer variável ou caracter de escape será expandido antes de ser atribuído.

Exemplo:

```
<?
$teste = "Brasil";
$php = '---$teste--\n';
echo "$php";
?>
```

A saída desse script será "---$teste--\n".

```
<?
$teste = "Brasil";
$php = "---$teste---\n";
echo "$php";
?>
```

A saída desse script será "---Brasil--" (com uma quebra de linha no final).

A tabela seguinte lista os caracteres de escape:

Sintaxe	Significado
\n	Nova linha
\r	Retorno de carro (semelhante a \n)
\t	Tabulação horizontal
\\	A própria barra (\ )
\$	O símbolo $
\’	Aspa simples
\”	Aspa dupla

Arrays

Arrays em PHP podem ser observados como mapeamentos ou como vetores indexados. Mais precisamente, um valor do tipo array é um dicionário onde os índices são as chaves de acesso. Vale ressaltar que os índices podem ser valores de qualquer tipo e não somente inteiros. Inclusive, se os índices forem todos inteiros, estes não precisam formar um intervalo contínuo

Como a checagem de tipos em PHP é dinâmica, valores de tipos diferentes podem ser usados como índices de array, assim como os valores mapeados também podem ser de diversos tipos.

```
Exemplo:
<?
$cor[1] = "vermelho";
$cor[2] = "verde";
$cor[3] = "azul";
$cor["teste"] = 1;
?>

Equivalentemente, pode-se escrever:

<?
$cor = array(1 => "vermelho, 2 => "verde, 3 => "azul",
"teste => 1);
?>
```

Listas

As listas são utilizadas em PHP para realizar atribuições múltiplas. Através de listas é possível atribuir valores que estão num array para variáveis. Vejamos o exemplo:

```
Exemplo:

list($a, $b, $c) = array("a", "b", "c");
```

O comando acima atribui valores às três variáveis simultaneamente. É bom notar que só são atribuídos às variáveis da lista os elementos do array que possuem índices inteiros e não negativos. No exemplo acima as três atribuições foram bem sucedidas porque ao inicializar um array sem especificar os índices eles passam a ser inteiros, a partir do zero. Um fator importante é que cada variável da lista possui um índice inteiro e ordinal, iniciando com zero, que serve para determinar qual valor será atribuído. No exemplo anterior temos $a com índice 0, $b com índice 1 e $c com índice 2. Vejamos um outro exemplo:

```
$arr = array(1=>"um",3=>"tres","a"=>"letraA",2=>"dois);
```

```
list($a,$b,$c,$d) = $arr;
```

Após a execução do código acima temos os seguintes valores:

```
$a == null
$b == "um"
$c == "dois"
$d == "tres"
```

Devemos observar que à variável $a não foi atribuído valor, pois no array não existe elemento com índice 0 (zero). Outro detalhe importante é que o valor "tres" foi atribuído à variável $d, e não a $b, pois seu índice é 3, o mesmo que $d na lista. Por fim, vemos que o valor "letraA" não foi atribuído a elemento algum da lista, pois seu índice não é inteiro.

Os índices da lista servem apenas como referência ao interpretador PHP para realizar as atribuições, não podendo ser acessados de maneira alguma pelo programador. De maneira diferente do array, uma lista não pode ser atribuída a uma variável, servindo apenas para fazer múltiplas atribuições através de um array.

Objetos

Um objeto pode ser inicializado utilizando o comando *new* para instanciar uma classe para uma variável.

```
Exemplo:
class teste {
     function nada() {
          echo "nada";
     }
}

$php = new teste;
$php -> nada();
```

A utilização de objetos será mais detalhada mais à frente.

Booleanos

PHP não possui um tipo booleano, mas é capaz de avaliar expressões e retornar *true* ou *false*, através do tipo `integer`: é usado o valor 0 (zero) para representar o estado *false*, e qualquer valor diferente de zero (geralmente 1) para representar o estado *true*.

Transformação de tipos

A transformação de tipos em PHP pode ser feita das seguintes maneiras:

Coerções

Quando ocorrem determinadas operações ("+", por exemplo) entre dois valores de tipos diferentes, o PHP converte o valor de um deles automaticamente (coerção). É interessante notar que se o operando for uma variável, seu valor não será alterado.

O tipo para o qual os valores dos operandos serão convertidos é determinado da seguinte forma: Se um dos operandos for `float`, o outro será convertido para `float`, senão, se um deles for `integer`, o outro será convertido para `integer`.

```
Exemplo:
$php = "1";           // $php é a string "1"
$php = $php + 1;      // $php é o integer 2
$php = $php + 3.7;// $php é o double 5.7
$php = 1 + 1.5        // $php é o double 2.5
```

Como podemos notar, o PHP converte `string` para `integer` ou `double` mantendo o valor. O sistema utilizado pelo PHP para converter de *strings* para números é o seguinte:

- É analisado o início da `string`. Se contiver um número, ele será avaliado. Senão, o valor será 0 (zero);
- O número pode conter um sinal no início ("+" ou "-");
- Se a `string` contiver um ponto em sua parte numérica a ser analisada, ele será considerado, e o valor obtido será `double`;
- Se a `string` contiver um "e" ou "E" em sua parte numérica a ser analisada, o valor seguinte será considerado como expoente da base 10, e o valor obtido será `double`;

```
Exemplos:
$php = 1 + "10.5";         // $php == 11.5
$php = 1 + "-1.3e3"; // $php == -1299
$php = 1 + "teste10.5";    // $php == 1
$php = 1 + "10testes";     // $php == 11
$php = 1 + "  10testes";   // $php == 11
$php = 1 + "+  10testes"; // $php == 1
```

Transformação explícita de tipos

A sintaxe do *typecast* de PHP é semelhante ao C: basta escrever o tipo entre parênteses antes do valor

```
Exemplo:
$php = 15;                  // $php é integer (15)
$php = (double) $php // $php é double (15.0)
$php = 3.9                  // $php é double (3.9)
$php = (int) $php    // $php é integer (3)
                           // o valor decimal é truncado
```

Os tipos de *cast* permitidos são:

(int), (integer)	⇒ muda para integer;
(real), (double), (float)	⇒ muda para float;
(string)	⇒ muda para string;
(array)	⇒ muda para array;
(object)	⇒ muda para objeto.

Com a função settype

A função `settype` converte uma variável para o tipo especificado, que pode ser "integer", "double", "string", "array" ou "object".

```
Exemplo:
$php = 15;                  // $php é integer
settype($php,double) // $php é double
```

4. Constantes

Constantes pré-definidas

O PHP possui algumas constantes pré-definidas, indicando a versão do PHP, o Sistema Operacional do servidor, o arquivo em execução, e diversas outras informações. Para ter acesso a todas as constantes pré-definidas, pode-se utilizar a função `phpinfo()`, que exibe uma tabela contendo todas as constantes pré-definidas, assim como configurações da máquina, sistema operacional, servidor http e versão do PHP instalada.

Definindo constantes

Para definir constantes utiliza-se a função `define`. Uma vez definido, o valor de uma constante não poderá mais ser alterado. Uma constante só pode conter valores escalares, ou seja, não pode conter nem um array nem um objeto. A assinatura da função `define` é a seguinte:

```
int define(string nome_da_constante, mixed valor);
```

A função retorna `true` se for bem-sucedida. Veja um exemplo de sua utilização a seguir:

```
define ("pi", 3.1415926536);
$circunf = 2*pi*$raio;
```

5. Operadores

Aritméticos

Só podem ser utilizados quando os operandos são números (integer ou float). Se forem de outro tipo, terão seus valores convertidos antes da realização da operação.

+	adição
-	subtração
*	multiplicação
/	divisão
%	módulo

de strings

Só há um operador exclusivo para strings:

.	concatenação

de atribuição

Existe um operador básico de atribuição e diversos derivados. Sempre retornam o valor atribuído. No caso dos operadores derivados de atribuição, a operação é feita entre os dois operandos, sendo atribuído o resultado para o primeiro. A atribuição é sempre por valor, e não por referência.

=	atribuição simples
+=	atribuição com adição
-=	atribuição com subtração
*=	atribuição com multiplicação
/=	atribuição com divisão
%=	atribuição com módulo
.=	atribuição com concatenação

```
Exemplo:

$a = 7;
$a += 2; // $a passa a conter o valor 9
```

bit a bit

Comparam dois números bit a bit.

&	"e" lógico
|	"ou" lógico
^	ou exclusivo
~	não (inversão)
<<	shift left
>>	shift right

Lógicos

Utilizados para inteiros representando valores booleanos

and	"e" lógico
or	"ou" lógico
xor	ou exclusivo
!	não (inversão)
&&	"e" lógico
||	"ou" lógico

Existem dois operadores para "e" e para "ou porque eles têm diferentes posições na ordem de precedência.

Comparação

As comparações são feitas entre os valores contidos nas variáveis, e não as referências. Sempre retornam um valor booleano.

==	igual a
!=	diferente de

<	menor que
>	maior que
<=	menor ou igual a
>=	maior ou igual a

Expressão condicional

Existe um operador de seleção que é ternário. Funciona assim:

```
(expressao1)?(expressao2):( expressao3)
```

o interpretador PHP avalia a primeira expressão. Se ela for verdadeira, a expressão retorna o valor de expressão2. Senão, retorna o valor de expressão3.

de incremento e decremento

++	incremento
--	decremento

Podem ser utilizados de duas formas: antes ou depois da variável. Quando utilizado antes, retorna o valor da variável antes de incrementá-la ou decrementá-la. Quando utilizado depois, retorna o valor da variável já incrementado ou decrementado.

```
Exemplos:

$a = $b = 10; // $a e $b recebem o valor 10
$c = $a++; // $c recebe 10 e $a passa a ter 11
$d = ++$b; // $d recebe 11, valor de $b já incrementado
```

Ordem de precedência dos operadores

A tabela a seguir mostra a ordem de precedência dos operadores no momento de avaliar as expressões;

Precedência	Associatividade	Operadores
1.	Esquerda	,
2.	Esquerda	or

Precedência	Associatividade	Operadores
3.	Esquerda	xor
4.	Esquerda	and
5.	Direita	print
6.	Esquerda	= += -= *= /= .= %= &= != ~= <<= >>=
7.	Esquerda	? :
8.	Esquerda	||
9.	Esquerda	&&
10.	Esquerda	|
11.	Esquerda	^
12.	Esquerda	&
13.	não associa	== !=
14.	não associa	< <= > >=
15.	Esquerda	<< >>
16.	Esquerda	+ - .
17.	Esquerda	* / %
18.	Direita	! ~ ++ -- (int) (double) (string) (array) (object) @
19.	Direita	[
20.	não associa	new

6. Estruturas de Controle

As estruturas que veremos a seguir são comuns para as linguagens de programação imperativas, bastando, portanto, descrever a sintaxe de cada uma delas, resumindo o funcionamento.

Blocos

Um bloco consiste de vários comandos agrupados com o objetivo de relacioná-los com determinado comando ou função. Em comandos como `if`, `for`, `while`, `switch` e em declarações de funções blocos podem ser utilizados para permitir que um comando faça parte do contexto desejado. Blocos em PHP são delimitados pelos caracteres "{" e "}". A utilização dos delimitadores de bloco em uma parte qualquer do código não relacionada com os comandos citados ou funções não produzirá efeito algum, e será tratada normalmente pelo interpretador.

```
Exemplo:
if ($x == $y)
     comando1;
comando2;
```

Para que comando2 esteja relacionado ao if é preciso utilizar um bloco:

```
if ($x == $y){
     comando1;
     comando2;
}
```

Comandos de seleção

Também chamados de condicionais, os comandos de seleção permitem executar comandos ou blocos de comandos com base em testes feitos durante a execução.

if

O mais trivial dos comandos condicionais é o `if`. Ele testa a condição e executa o comando indicado se o resultado for `true` (valor diferente de zero). Ele possui duas sintaxes:

```
if (expressão)
     comando;

if (expressão){
     comando1;
     comando2;
```

```
     comando3;
     comando4;
     comando5;
     comando1;
     comando2;
     comando3;
}
```

```
if (expressão):
     comando;
. . .
     comando;
endif;
```

Para incluir mais de um comando no `if` da primeira sintaxe, é preciso utilizar um bloco, demarcado por chaves.

O `else` é um complemento opcional para o `if`. Se utilizado, o comando será executado se a expressão retornar o valor `false` (zero). Suas duas sintaxes são:

```
if (expressão)
comando;
else
comando;
```

```
if (expressão):
     comando;
. . .
     comando;
else
     comando;
. . .
     comando;
endif;
```

A seguir, temos um exemplo do comando `if` utilizado com `else`:

```
if ($a > $b)
     $maior = $a;
else
     $maior = $b;
```

O exemplo acima coloca em $maior o maior valor entre $a e $b

Em determinadas situações é necessário fazer mais de um teste, e executar condicionalmente diversos comandos ou blocos de comandos. Para facilitar o entendimento de uma estrutura do tipo:

```
if (expressao1)
     comando1;
else
     if (expressao2)
          comando2;
     else
          if (expressao3)
               comando3;
          else
               comando4;
```

foi criado o comando, também opcional `elseif`. Ele tem a mesma função de um `else` e um `if` usados seqüencialmente, como no exemplo acima. Num mesmo `if` podem ser utilizados diversos `elseif`'s, ficando essa utilização a critério do programador, que deve zelar pela legibilidade de seu script.

O comando `elseif` também pode ser utilizado com dois tipos de sintaxe. Em resumo, a sintaxe geral do comando `if` fica das seguintes maneiras:

```
if (expressao1)
     comando;
[ elseif (expressao2)
     comando; ]
[ else
     comando; ]
```

```
if (expressao1) :
    comando;
. . .
    comando;
[ elseif (expressao2)
    comando;
. . .
    comando; ]
[ else
    comando;
. . .
    comando; ]
endif;
```

switch

O comando `switch` atua de maneira semelhante a uma série de comandos `if` na mesma expressão. Freqüentemente o programador pode querer comparar uma variável com diversos valores, e executar um código diferente a depender de qual valor é igual ao da variável. Quando isso for necessário, deve-se usar o comando `switch`. O exemplo seguinte mostra dois trechos de código que fazem a mesma coisa, sendo que o primeiro utiliza uma série de `if`'s e o segundo utiliza `switch`:

```
if ($i == 0)
    print "i é igual a zero";
elseif ($i == 1)
    print "i é igual a um";
elseif ($i == 2)
    print "i é igual a dois";

switch ($i) {
    case 0:
        print "i é igual a zero";
    break;
    case 1:
        print "i é igual a um";
    break;
    case 2:
        print "i é igual a dois";
    break;
}
```

É importante compreender o funcionamento do `switch` para não cometer enganos. O comando `switch` testa linha a linha os cases encontrados, e a partir do momento que encontra um valor igual ao da variável testada, passa a executar todos os comandos seguintes, mesmo os que fazem parte de

outro teste, até o fim do bloco. por isso usa-se o comando `break`, quebrando o fluxo e fazendo com que o código seja executado da maneira desejada. Veremos mais sobre o `break` mais adiante. Veja o exemplo:

```
switch ($i) {
     case 0:
          print "i é igual a zero";
     case 1:
          print "i é igual a um";
     case 2:
          print "i é igual a dois";
}
```

No exemplo acima, se $i for igual a zero, os três comandos "print" serão executados. Se $i for igual a 1, os dois últimos "print" serão executados. O comando só funcionará da maneira desejada se $i for igual a 2.

Em outras linguagens que implementam o comando switch, ou similar, os valores a serem testados só podem ser do tipo inteiro. Em PHP é permitido usar valores do tipo string como elementos de teste do comando switch. O exemplo abaixo funciona perfeitamente:

```
switch ($s) {
     case "casa":
          print "A casa é amarela";
     case "arvore":
          print "a árvore é bonita";
     case "lâmpada":
          print "João apagou a lâmpada";
    }
```

comandos de repetição

while

O `while` é o comando de repetição (laço) mais simples. Ele testa uma condição e executa um comando, ou um bloco de comandos, até que a condição testada seja falsa. Assim como o `if`, o `while` também possui duas sintaxes alternativas:

```
while (<expressão>)
     <comando>;

while (<expressão>){
     <comando1>;
```

```
        <comando2>;
        <comando3>;
        <comando4>;
        <comando5>;
        <comando3>;
}
```

```
while (<expressão>):
        <comando>;
. . .
        <comando>;
endwhile;
```

A expressão só é testada a cada vez que o bloco de instruções termina, além do teste inicial. Se o valor da expressão passar a ser `false` no meio do bloco de instruções, a execução segue até o final do bloco. Se no teste inicial a condição for avaliada como `false`, o bloco de comandos não será executado.

O exemplo a seguir mostra o uso do `while` para imprimir os números de 1 a 10:

```
$i = 1;
while ($i <=10)
print $i++;
```

do... while

O laço `do...while` funciona de maneira bastante semelhante ao `while`, com a simples diferença que a expressão é testada ao final do bloco de comandos. O laço `do...while` possui apenas uma sintaxe, que é a seguinte:

```
do {
        <comando>
. . .
        <comando>
} while (<expressão>);
```

O exemplo utilizado para ilustrar o uso do `while` pode ser feito da seguinte maneira utilizando o `do... while`:

```
$i = 0;
do {
print ++$i;
} while ($i < 10);
```

for

O tipo de laço mais complexo é o `for`. Para os que programam em C, C++ ou Java, a assimilação do funcionamento do for é natural. Mas para aqueles que estão acostumados a linguagens como Pascal, há uma grande mudança para o uso do for. As três sintaxes permitidas são:

```
for (<inicialização>;<condição>;<incremento ou decremento>)
     <comando>;

for (<inicialização>;<condição>;<incremento ou decremento>){
     <comando>;
     <comando>;
     <comando>;
     <comando>;
     <comando>;
}

for (<inicialização>;<condição>;<incremento ou decremento>):
     <comando>;
. . .
     <comando>;
endfor;
```

As três expressões que ficam entre parênteses têm as seguintes finalidades:

Inicialização: comando ou seqüência de comandos a serem realizados antes do inicio do laço. Serve para inicializar variáveis.

Condição: Expressão booleana que define se os comandos que estão dentro do laço serão executados ou não. Enquanto a expressão for verdadeira (valor diferente de zero) os comandos serão executados.

Incremento: Comando executado ao final de cada execução do laço.

Um comando `for` funciona de maneira semelhante a um `while` escrito da seguinte forma:

```
<inicialização>
while (<condição>) {
     comandos
...
     <incremento>
}
```

Quebra de fluxo

Break

O comando `break` pode ser utilizado em laços de `do`, `for` e `while`, além do uso já visto no comando `switch`. Ao encontrar um `break` dentro de um desses laços, o interpretador PHP para imediatamente a execução do laço, seguindo normalmente o fluxo do script.

```
while ($x > 0) {
...
     if ($x == 20) {
          echo "erro! x = 20";
     break;
...
}
```

No trecho de código acima, o laço `while` tem uma condição para seu término normal ($x <= 0), mas foi utilizado o break para o caso de um término não previsto no início do laço. Assim o interpretador seguirá para o comando seguinte ao laço.

Continue

O comando `continue` também deve ser utilizado no interior de laços, e funciona de maneira semelhante ao `break`, com a diferença que o fluxo ao invés de sair do laço volta para o início dele. Vejamos o exemplo:

```
for ($i = 0; $i < 100; $i++) {
     if ($i % 2) continue;
          echo " $i ";
}
```

O exemplo acima é uma maneira ineficiente de imprimir os números pares entre 0 e 99. O que o laço faz é testar se o resto da divisão entre o número e 2 é 0. Se for diferente de zero (valor lógico `true`) o interpretador encontrará um `continue`, que faz com que os comandos seguintes do interior do

laço sejam ignorados, seguindo para a próxima iteração.

7. Funções

Definindo funções

A sintaxe básica para definir uma função é:

```
function nome_da_função([arg1, arg2, arg3]) {
     Comandos;
... ;
     [return <valor de retorno>];
}
```

Qualquer código PHP válido pode estar contido no interior de uma função. Como a checagem de tipos em PHP é dinâmica, o tipo de retorno não deve ser declarado, sendo necessário que o programador esteja atento para que a função retorne o tipo desejado. É recomendável que esteja tudo bem documentado para facilitar a leitura e compreensão do código. Para efeito de documentação, utiliza-se o seguinte formato de declaração de função:

```
tipo function nome_da_funcao(tipo arg1, tipo arg2, ...);
```

Este formato só deve ser utilizado na documentação do script, pois o PHP não aceita a declaração de tipos. Isso significa que em muitos casos o programador deve estar atento ao tipos dos valores passados como parâmetros, pois se não for passado o tipo esperado não é emitido nenhum alerta pelo interpretador PHP, já que este não testa os tipos.

Valor de retorno

Toda função pode opcionalmente retornar um valor, ou simplesmente executar os comandos e não retornar valor algum.

Não é possível que uma função retorne mais de um valor, mas é permitido fazer com que uma função retorne um valor composto, como listas ou arrays.

Argumentos

É possível passar argumentos para uma função. Eles devem ser declarados logo após o nome da função, entre parênteses, e tornam-se variáveis pertencentes ao escopo local da função. A declaração do

tipo de cada argumento também é utilizada apenas para efeito de documentação.

```
Exemplo:

function imprime($texto){
     echo $texto;
}

imprime("teste de funções");
```

Passagem de parâmetros por referência

Normalmente, a passagem de parâmetros em PHP é feita por valor, ou seja, se o conteúdo da variável for alterado, essa alteração não afeta a variável original.

```
Exemplo:

function mais5($numero) {
$numero += 5;
}

$a = 3;
mais5($a); //$a continua valendo 3
```

No exemplo acima, como a passagem de parâmetros é por valor, a função mais5 é inútil, já que após a execução sair da função o valor anterior da variável é recuperado. Se a passagem de valor fosse feita por referência, a variável $a teria 8 como valor. O que ocorre normalmente é que ao ser chamada uma função, o interpretador salva todo o escopo atual, ou seja, os conteúdos das variáveis. Se uma dessas variáveis for passada como parâmetro, seu conteúdo fica preservado, pois a função irá trabalhar na verdade com uma cópia da variável.7 Porém, se a passagem de parâmetros for feita por referência, toda alteração que a função realizar no valor passado como parâmetro afetará a variável que o contém.

Há duas maneiras de fazer com que uma função tenha parâmetros passados por referência: indicando isso na declaração da função, o que faz com que a passagem de parâmetros sempre seja assim; e também na própria chamada da função. Nos dois casos utiliza-se o modificador "&". Vejamos um exemplo que ilustra os dois casos:

```
function mais5(&$num1, $num2) {
$num1 += 5;
$num2 += 5;
}
```

```
$a = $b = 1;
mais5($a, $b); /* Neste caso, só $num1 terá seu valor alterado, pois a passagem por referência está definida na declaração da função. */


mais5($a, &$b); /* Aqui as duas variáveis terão seus valores alterados. */
```

Argumentos com valores pré-definidos (default)

Em PHP é possível ter valores *default* para argumentos de funções, ou seja, valores que serão assumidos em caso de nada ser passado no lugar do argumento. Quando algum parâmetro é declarado desta maneira, a passagem do mesmo na chamada da função torna-se opcional.

```
function teste($php = "testando") {
    echo $php;
}

teste(); // imprime "testando"
teste("outro teste"); // imprime "outro teste"
```

É bom lembrar que quando a função tem mais de um parâmetro, o que tem valor *default* deve ser declarado por último:

```
function teste($figura = circulo, $cor) {
    echo "a figura é um ". $figura. " de cor " $cor;
}

teste(azul);
/* A função não vai funcionar da maneira esperada, ocorrendo um erro no interpretador. A declaração correta é: */

function teste2($cor, $figura = circulo) {
    echo "a figura é um ". $figura. " de cor " $cor;
}

teste2(azul);

/* Aqui a função funciona da maneira esperada, ou seja, imprime o texto: "a figura é um círculo de cor azul" */
```

Contexto

O contexto é o conjunto de variáveis e seus respectivos valores num determinado ponto do programa. Na chamada de uma função, ao iniciar a execução do bloco que contém a implementação da mesma é criado um novo contexto, contendo as variáveis declaradas dentro do bloco, ou seja, todas as variáveis utilizadas dentro daquele bloco serão eliminadas ao término da execução da função.

Escopo

O escopo de uma variável em PHP define a porção do programa onde ela pode ser utilizada. Na maioria dos casos todas as variáveis têm escopo global. Entretanto, em funções definidas pelo usuário um escopo local é criado. Uma variável de escopo global não pode ser utilizada no interior de uma função sem que haja uma declaração.

```
Exemplo:
$php = "Testando";

function Teste() {
echo $php;
}

Teste();
```

O trecho acima não produzirá saída alguma, pois a variável $php é de escopo global, e não pode ser referida num escopo local, mesmo que não haja outra com nome igual que cubra a sua visibilidade. Para que o script funcione da forma desejada, a variável global a ser utilizada deve ser declarada.

```
Exemplo:
$php = "Testando";

function Teste() {
global $php;
echo $php;
}

Teste();
```

Uma declaração "global" pode conter várias variáveis, separadas por vírgulas. Uma outra maneira de acessar variáveis de escopo global dentro de uma função é utilizando um array pré-definido pelo PHP cujo nome é $GLOBALS. O índice para a variável referida é o próprio nome da variável, sem o caracter $. O exemplo acima e o abaixo produzem o mesmo resultado:

```
Exemplo:
$php = "Testando";
```

```
function Teste() {
echo $GLOBALS["php"]; // imprime $php
echo $php; // não imprime nada
}

Teste();
```

8. Variáveis

O modificador static

Uma variável estática é visível num escopo local, mas ela é inicializada apenas uma vez e seu valor não é perdido quando a execução do script deixa esse escopo. Veja o seguinte exemplo:

```
function Teste() {
$a = 0;
echo $a;
$a++;
}
```

O último comando da função é inútil, pois assim que for encerrada a execução da função a variável $a perde seu valor. Já no exemplo seguinte, a cada chamada da função a variável $a terá seu valor impresso e será incrementada:

```
function Teste() {
static $a = 0;
echo $a;
$a++;
}
```

O modificador static é muito utilizado em funções recursivas, já que o valor de algumas variáveis precisa ser mantido. Ele funciona da seguinte forma: O valor das variáveis declaradas como estáticas é mantido ao terminar a execução da função. Na próxima execução da função, ao encontrar novamente a declaração com static, o valor da variável é recuperado.

Em outras palavras, uma variável declarada como `static` tem o mesmo "tempo de vida" que uma variável global, porém sua visibilidade é restrita ao escopo local em que foi declarada e só é recuperada após a declaração.

Exemplo:

```
function Teste() {
echo "$a";
static $a = 0;
$a++;
}
```

O exemplo acima não produzirá saída alguma. Na primeira execução da função, a impressão ocorre antes da atribuição de um valor à função e, portanto o conteúdo de $a é nulo (string vazia). Nas execuções seguintes da função Teste() a impressão ocorre antes da recuperação do valor de `$a` e, portanto

nesse momento seu valor ainda é nulo. Para que a função retorne algum valor o modificador `static` deve ser utilizado.

Variáveis Variáveis

O PHP tem um recurso conhecido como variáveis variáveis, que consiste em variáveis cujos nomes também são variáveis. Sua utilização é feita através do duplo cifrão ($$).

```
$a = "teste";
$$a = "Mauricio Vivas";

O exemplo acima e equivalente ao seguinte:

$a = "teste";
$teste = "Mauricio Vivas";
```

Variáveis enviadas pelo navegador

Para interagir com a navegação feita pelo usuário, é necessário que o PHP possa enviar e receber informações para o software de navegação. A maneira de enviar informações, como já foi visto anteriormente, geralmente é através de um comando de impressão, como o *echo*. Para receber informações vindas do navegador através de um *link* ou um formulário html o PHP utiliza as informações enviadas através da URL. Por exemplo: se seu script php está localizado em "`http://localhost/teste.php3`" e você o chama com a url "`http://localhost/teste.php3?php=teste`", automaticamente o PHP criará uma variável com o nome $php contendo a string "teste". Note que o conteúdo da variável está no formato `urlencode`. Os formulários html já enviam informações automaticamente nesse formato, e o PHP decodifica sem necessitar de tratamento pelo programador.

URLencode

O formato `urlencode` é obtido substituindo os espaços pelo caracter "+" e todos os outros caracteres não alfa-numéricos (com exceção de "_") pelo caracter "%" seguido do código ASCII em hexadecimal.

Por exemplo: o texto "Testando 1 2 3 !!" em urlencode fica "Testando+1+2+3+%21%21"

O PHP possui duas funções para tratar com texto em urlencode. Seguem suas sintaxes:

```
string urlencode(string texto);
string urldecode(string texto);
```

Essas funções servem respectivamente para codificar ou decodificar um texto passado como argumento. Para entender melhor o que é um argumento e como funciona uma função, leia o tópico "funções".

Variáveis de ambiente

O PHP possui diversas variáveis de ambiente, como a `$PHP_SELF`, por exemplo, que contém o nome e o path do próprio arquivo. Algumas outras contém informações sobre o navegador do usuário, o servidor http, a versão do PHP e diversas informações. Para ter uma listagem de todas as variáveis e constantes de ambiente e seus respectivos conteúdos, deve-se utilizar a função `phpinfo()`.

Verificando o tipo de uma variável

Por causa da tipagem dinâmica utilizada pelo PHP, nem sempre é possível saber qual o tipo de uma variável em determinado instantes não contar com a ajuda de algumas funções que ajudam a verificar isso. A verificação pode ser feita de duas maneiras:

Função que retorna o tipo da variável

Esta função é a `gettype`. Sua assinatura é a seguinte:

```
string gettype(mixed var);
```

A palavra "mixed" indica que a variável var pode ser de diversos tipos.

A função `gettype` pode retornar as seguintes strings: "`integer`", "`double`", "`string`", "`array`", "`object`" e "`unknown type`".

Funções que testam o tipo da variável

São as funções `is_int`, `is_integer`, `is_real`, `is_long`, `is_float`,

`is_string`, `is_array` e `is_object`. Todas têm o mesmo formato, seguindo modelo da assinatura a seguir:

```
int is_integer(mixed var);
```

Todas essas funções retornam `true` se a variável for daquele tipo, e `false` em caso contrário.

Destruindo uma variável

É possível desalocar uma variável se ela não for usada posteriormente através da função `unset`, que tem a seguinte assinatura:

```
int unset(mixed var);
```

A função destrói a variável, ou seja, libera a memória ocupada por ela, fazendo com que ela deixe de existir. Se mais na frente for feita uma chamada á variável, será criada uma nova variável de mesmo nome e de conteúdo vazio, a não ser que a chamada seja pela função `isset`. Se a operação for bem sucedida, retorna `true`.

Verificando se uma variável possui um valor

Existem dois tipos de teste que podem ser feitos para verificar se uma variável está setada: com a função `isset` e com a função `empty`.

A função isset

Possui o seguinte protótipo:

```
int isset(mixed var);
```

E retorna `true` se a variável estiver setada (ainda que com uma string vazia ou o valor zero), e `false` em caso contrário.

A função empty

Possui a seguinte assinatura:

```
int empty(mixed var);
```

E retorna `true` se a variável não contiver um valor (não estiver setada) ou possuir valor 0 (zero) ou uma string vazia. Caso contrário, retorna `false`.

Arrays Multidimensionais

Arrays multidimensionais são arrays simples com um dos (ou todos) seus elementos sendo outro array e assim consecutivamente.
Exemplo:

```
$Campeao[5] = 123456789 ;

$Tricampeao["casa"] = $Campeao;
$Tricampeao["predio"] = 19191919;

$Brasil[1] = $Tricampeao;
$Brasil[2] = "Bicampeao";
$Brasil["copa"] = $Tricampeao;
$Brasil[4] = "Tetracampeao";
$Brasil["mundo"] = "Pentacampeao";


echo $Campeao[5];                  // resultará 123456789   Array simples
echo $Brasil[1]["casa"][5] ;       // resultará 19191919    Array tridimensional
echo $Tricampeao["casa"][5];       // resultará 123456789   Array bidimensional
echo $Brasil["copa"]["predio"];    // resultará 19191919    Array bidimensional
```

9. Classes e Objetos

Classe

Uma classe é um conjunto de variáveis e funções relacionadas a essas variáveis. Uma vantagem da utilização é poder usufruir o recurso de encapsulamento de informação. Com o encapsulamento o usuário de uma classe não precisa saber como ela é implementada, bastando para a utilização conhecer a interface, ou seja, as funções disponíveis. Uma classe é um tipo e, portanto não pode ser atribuída a uma variável. Para definir uma classe, deve-se utilizar a seguinte sintaxe:

```
class Nome_da_classe {
     var $variavel1;
     var $variavel2;
     function funcao1 ($parâmetro) {
          /* === corpo da função === */
     }
}
```

Objeto

Como foi dito anteriormente, classes são tipos, e não podem ser atribuídas a variáveis. Variáveis do tipo de uma classe são chamadas de objetos, e devem ser criadas utilizando o operador `new`, seguindo o exemplo abaixo:

```
$variável = new $nome_da_classe;
```

Para utilizar as funções definidas na classe, deve ser utilizado o operador "`->`", como no exemplo:

```
$variável->funcao1();
```

A variável $this

Na definição de uma classe, pode-se utilizar a variável `$this`, que é o próprio objeto. Assim, quando uma classe é instanciada em um objeto, e uma função desse objeto na definição da classe utiliza a variável `$this`, essa variável significa o objeto que estamos utilizando.

Como exemplo da utilização de classes e objetos, podemos utilizar a classe conta, que define uma conta bancária bastante simples, com funções para ver saldo e fazer um crédito.

```
class conta {
var $saldo;
function saldo() {
```

```
return $this->saldo;
}
function credito($valor) {
$this->saldo += $valor;
}
}

$minhaconta = new conta;
$minhaconta->saldo();     // a variável interna não foi
                          // inicializada, e não contém
                          // valor algum
$minhaconta->credito(50);
$minhaconta->saldo(); // retorna 50
```

SubClasses

Uma classe pode ser uma extensão de outra. Isso significa que ela herdará todas as variáveis e funções da outra classe, e ainda terá as que forem adicionadas pelo programador. Em PHP não é permitido utilizar herança múltipla, ou seja, uma classe pode ser extensão de apenas uma outra.Para criar uma classe extendida, ou derivada de outra, deve ser utilizada a palavra reservada `extends`, como pode ser visto no exemplo seguinte:

```
class novaconta extends conta {
     var $numero;
     function numero() {
          return $this->numero;
     }
}
```

A classe acima é derivada da classe conta, tendo as mesmas funções e variáveis, com a adição da variável `$numero` e a função `numero()`.

Construtores

Um construtor é uma função definida na classe que é automaticamente chamada no momento em que a classe é instanciada (através do operador `new`). O construtor deve ter o mesmo nome que a classe a que pertence. Veja o exemplo:

```
class conta {
     var $saldo;

     function conta () {
          $this.saldo = 0;
     }

     function saldo() {
          return $this->saldo;
```

```
        }
        function credito($valor) {
              $this->saldo += $valor;
        }
}
```

Podemos perceber que a classe conta agora possui um construtor, que inicializa a variável `$saldo` com o valor 0.

Um construtor pode conter argumentos, que são opcionais, o que torna esta ferramenta mais poderosa. No exemplo acima, o construtor da classe conta pode receber como argumento um valor, que seria o valor inicial da conta.

Vale observar que para classes derivadas, o construtor da classe pai não é automaticamente herdado quando o construtor da classe derivada é chamado.

10. PHP avançado

Interagindo com o browser

PHP também permite interagir com informações do browser automaticamente. Por exemplo, o script a seguir mostra informações sobre o browser do usuário.

```
<html>
<head><title>Aprendendo PHP</title></head>
<body>

<? echo $HTTP_USER_AGENT; ?>

</body>
</html>
```

Esse código em um Internet Explorer 6.0 com sistema operacional Windows 98, geraria: *Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)*

```
<html>
<head><title>Aprendendo PHP</title></head>
<body>

<?
if  (strpos($HTTP_USER_AGENT,"MSIE”) != 0) {
echo "Você usa Internet Explorer";
} else {
echo "Você não usa Internet Explorer";
}
?>

</body>
</html>
```

Neste exemplo, será apenas exibido um texto informando se está sendo utilizado o Microsoft Internet Explorer ou não, mas para outras funções poderia ser utilizado algo semelhante.

É bom notar o surgimento de mais uma função no código anterior: `strpos(string1,string2)`. Essa função retorna a posição da primeira aparição de string2 em string1, contando a partir de zero, e não retorna valor algum se não ocorrer. Assim, para testar se a string `$HTTP_USER_AGENT` contém a string “`MSIE`”, basta testar se `strpos` devolve algum valor.

Utilizando formulários HTML

Ao clicar num botão “Submit” em um formulário HTML as informações dos campos serão

enviadas ao servidor especificado para que possa ser produzida uma resposta. O PHP trata esses valores como variáveis, cujo nome é o nome do campo definido no formulário. O exemplo a seguir mostra isso, e mostra também como o código PHP pode ser inserido em **qualquer** parte do código HTML:

```
<html>
<head><title>Aprendendo PHP</title></head>
<body>

<?php
if ($texto != "")
echo "Você digitou \"$texto\"<br><br>";
?>

<form method=post action="<? echo $PATH_INFO; ?>">
<input type="text" name="texto" value="" size=10>
<br>
<input type="submit" name="sub" value="Enviar!">
</form>

</body>
</html>
```

Ao salvar o arquivo acima e carregá-lo no browser, o usuário verá apenas um formulário que contém um espaço para digitar o texto. Ao digitar um texto qualquer e submeter o formulário, a resposta, que é o mesmo arquivo PHP (indicado pela constante $PATH_INFO, que retorna o nome do arquivo) exibirá a mensagem "Você digitou <<mensagem>>".

Isso ocorre porque o código PHP testa o conteúdo da variável $texto. Inicialmente ele é uma string vazia, e por isso nada é impresso na primeira parte. Quando algum texto é digitado no formulário e submetido, o PHP passa a tratá-lo como uma variável. Como no formulário o campo possui o nome “texto”, a variável com seu conteúdo será $texto. Assim, no próximo teste o valor da variável será diferente de uma string vazia, e o PHP imprime um texto antes do formulário.

Cookies

Cookies são mecanismos para armazenar e consultar informações nos browsers dos visitantes da página. O PHP atribui cookies utilizando a função **setcookie**, que deve ser utilizada antes da tag <html> numa página.

O uso de cookies não é recomendado quando se trata de informações sigilosas. Os dados dos cookies são armazenados no diretório de arquivos temporários do visitante, sendo facilmente visualizado por pessoas mal intencionadas.

Além da opção "aceitar cookies" que pode ser desativada a qualquer momento pelo visitante.

Para uma transmissão de dados segura é recomendável o uso de sessões(ver adiante).

Setcookie("nome_do_cookie","seu_valor","tempo_de_vida","path","domínio","conexão_segura")

Nome_do_cookie = É o nome que, posteriormente, se tornará a variável e o que o servirá de referência para indicar o cookie.
Seu_valor = É o valor que a variável possuirá. Esse valor pode ser de todos os tipos.
Tempo_de_vida = É o tempo, em segundos, que o cookie existirá no computador do visitante. Uma vez excedido esse prazo o cookie se apaga de modo irrecuperável. Se esse argumento ficar vazio, o cookie se apagará quando o visitante fechar o browser.
Path = endereço da página que gerou o cookie – automático
Domínio = domínio ao qual pertence o cookie – automático
Conexão_segura = Indica se o cookie deverá ser transmitido somente em uma conexão segura HTTPS.

Sessão

Sessões são mecanismos muito parecidos com os tradicionais cookies. Suas diferenças são que sessões são armazenadas no próprio servidor e não expiram a menos que o programador queira apagar a sessão.

Existem algumas funções que controlam sessões e estão detalhadas no capítulo "12.Bibliotecas de funções".

Aqui estão as funções de sessão mais usadas.

Nome da função	Argumentos
Session_start()	Não precisa de argumento
Session_register()	A variável sem o cifrão
Session_unregister()	A variável sem o cifrão
Session_is_registered()	A variável sem o cifrão

O *session_destroy()* só deve ser usado quando for da vontade do programador acabar com todos as sessões daquele visitante, portanto muito cuidado com essa função.

Require

A função **require** põe o conteúdo de um outro arquivo no arquivo php atual, antes de ser executado. Quando o interpretador do PHP ler este arquivo, ele encontrará todo o conteúdo dos “require´s” adicionado no arquivo corrente.

```
Require(“nomedoarquivo”);
```

Criando o hábito de usar essa função, o programador pode vir a encontrar um erro de arquivo já declarado. Para evitar isso é recomendável que sempre que a função **require** for utilizada ela seja substituída pela função **require_once**.

```
Require_once(“nome_do_arquivo“);
```

Include

A função **Include** é semelhante à função **require**, com a diferença que o código do arquivo incluído é processado em tempo de execução, permitindo que sejam usados “includes” dentro de estruturas de controle como **for** e **while**.

```
$arqs = array(‘a1.inc’,’a2.inc’,’a3.inc’) ;
for ($i=0;$i<count($arqs); $i++){
        include($arqs[$i]);
}
if ($x == $y){
        include($arquivo1);
}
else{
        include($arquivo2);
}
/*
        Note que quando se utiliza a função include
Dentro de estruturas é necessário a utilização das chaves
*/
```

Formulários Avançados

O PHP também entende as matrizes em forma de variáveis vindas de um formulário.

Exemplo:

```
<form action=”matrizes.php” method=“POST”>
        Nome : <input type=”text” name=dados[nome]><br>
        Email : <input type=”text” name=dados[email]><br>
        Esportes de preferência: <br>
```

```
<select multiple name="esportes[]">
        <option value="futebol">Futebol</option>
        <option value="vôlei">Vôlei</option>
</select><br>
<input type="submit" value="enviar">

</form>
```

Depois de enviado, os campos se tornarão variáveis e campos como este "dados[nome]" se transformarão em arrays que não deixam de ser variáveis.

Arquivos Remotos

Nas funções que trabalham com arquivos, é possível utilizar URLs para acessar arquivos em outros servidores na Web ou no próprio servidor.

As funções aqui utilizadas serão detalhadas mais à frente no capítulo "Biblioteca de Funções".

Exemplo:

Lendo

```
$f = fopen("http://www.php.net","r");
if (!$f){
        echo "Erro ao abrir a URL.<br>";
        exit;
}
while (!feof($f)){
        $s = fgets($f,256);
        echo $s;
}
fclose($f);
```

Escrevendo

```
$f = fopen("ftp://user:senha@site","w");
if (!$f){
        echo "Erro ao abrir a URL.<br>";
        exit;
}
else{
        fputs($f,"texto a ser escrito");
        fputs($f,"mais texto a ser escrito");
        fputs($f,"mais texto a ser escrito");
        fputs($f,"mais texto a ser escrito");
        fputs($f,"mais texto a ser escrito");
        fclose;
```

}

Tratamento de erros

Existem quatro tipos(até a versão 4.0) de erros no PHP para indicar a gravidade do erro encontrado ou ocorrido. Eles são:

1. Erros de funções (function errors)
2. Avisos (warnings)
3. Erros de processamento (parser error)
4. Observações (notice)

As mensagens de erro são uma coisa com que os programadores devem prestar muita atenção, afinal nenhum programador quer por no ar um sistema que quando o primeiro visitante entra apareça uma mensagem de erro. Para evitar essas inconveniências use sempre um "@" antes da cada chamada as funções. Se a opção **track_errors** no arquivo ***php.ini*** estiver habilitada, a mensagem de erro poderá ser encontrada na variável global **$php_errormsg**.

A chamada da função ficaria assim:

```
@strtolower();
```

Essa função deixaria todos os caracteres em minúsculo, mas como não foi passado nenhum argumento essa função deveria exibir uma mensagem de erro.

11.SQL

O que é?

De acordo com Andrew Taylor, inventor original da linguagem SQL, o SQL não significa "Structured Query language" (ou qualquer outra coisa nesse sentido). Mas para o resto do mundo, esse é seu significado agora. Como você poderia esperar desse (não-)título, o SQL representa um método mais geral e mais estrito de armazenamento de dados que o padrão anterior de banco de dados não-relacionais no estilo dbm.

SQL não é ciência espacial. As quatro instruções básicas de manipulação de dados suportadas essencialmente por todos os bancos de dados de SQL são SELECT, INSERT, UPDATE e DELETE. SELECT recupera os dados do banco de dados, INSERT insere em uma nova entrada, UPDATE edita partes da entrada no lugar e DELETE exclui uma entrada completamente.

A maior parte da dificuldade reside em projetar bancos de dados. O projetista deve pensar muito seriamente sobre a melhor maneira de representar cada parte de dados e relacionamento para a utilização planejada. É um prazer programar com bancos de dados bem projetados, enquanto os pobremente projetados podem deixar você careca quando contemplar as numerosas conexões e horrorosas junções.

Os bancos de dados SQL são criados pelas chamadas instruções de estrutura de dados. As mais importantes são CREATE, ALTER e DROP. Como se poderia imaginar, CREATE DATABASE cria um novo banco de dados e CREATE TABLE define uma nova tabela dentro de um banco de dados. ALTER altera a estrutura de uma tabela. DROP é a bomba nuclear entre os comandos de SQL, uma vez que exclui completamente tabelas ou banco de dados inteiros.

O bom projeto de banco de dados também é uma questão de segurança. Empregando medidas profiláticas razoáveis, um banco de dados SQL pode aprimorar a segurança de seu site.

Esta apostila explicará somente o básico de SQL.

SELECT

SELECT é o comando principal de que você precisa para obter informações fora de um banco de dados de SQL. A sintaxe básica é extremamente simples:

SELECT campo1, campo2, campo3 FROM tabela WHERE condições

Em alguns casos, você desejará pedir linhas inteiras em vez de selecionar partes individuais de informações. Essa prática é obsoleta por razões muito boas(ela pode ser mais lenta que solicitar apenas os dados de que você precisa e pode levar a problemas se você reprojetar a tabela) , mas ainda é utilizada e, portanto, precisamos menciona-la. Uma linha inteira é indicada utilizando o símbolo do asteristico:

SELECT * FROM minha_tabela WHERE campo1 = "abcdef"

Subseleções

Antes de deixarmos o reino das instruções SELECT, devemos mencionar a subseleção. Essa é uma instrução como:

SELECT phone_number FROM table WHERE name = 'SELECT name FROM table2 WHERE id = 1';

As subseleções são mais uma conveniência que uma necessidade. Elas podem ser muito úteis se você estiver trabalhando com enormes lotes de dados; mas é possível obter o mesmo resultado com duas seleções mais simples(embora isso será algo mais lento, mesmo no PHP4).

INSERT

O comando que você precisa para colocar novos dados em um banco de dados é o INSERT. Eis a sintaxe básica:

INSERT INTO tabela (campo1,campo2,campo3) VALUES($val1,$val2,$val3);

Obviamente as colunas e seus valores devem corresponder; se você misturar os seus itens de array, nada de bom acontecerá. Se algumas linhas não tiverem valores para alguns campos, você precisará utilizar um valor nulo, vazio ou auto-incrementado – e, em um nível mais profundo, você pode ter assegurado de antemão que os campos possam ser nulos ou auto-incrementáveis. Se isso não for possível, você simplesmente deve omitir qualquer coluna que você deseja que assuma o padrão de valor vazio em uma instrução de INSERT.

Um desdobramento da instrução INSERT básica é INSERT INTO... SELECT. Isso apenas significa que você pode inserir os resultados de uma instrução SELECT:

INSERT INTO customer(birthmonth, birthflower, birthstone) SELECT * FROM birthday_info WHERE birthmonth = $birthmonth;

Entretanto, nem todos os bancos de dados de SQL têm essa capacidade. Além disso, é preciso ser cuidadoso com esse comando porque você pode facilmente causar problemas para você mesmo. Em geral, não é uma boa idéia selecionar a partir do mesmo banco de dados em que você esta inserindo.

UPDATE

UPDATE é utilizado para editar informações que já estão no banco de dados sem excluir qualquer quantidade significativa. Em outras palavras, você pode seletivamente alterar algumas informações sem excluir um registro antigo inteiro e inserir um novo. A sintaxe é:

UPDATE table SET campo1 = “val1”, campo2 = “val2”,campo3=”val3” WHERE condição;

A instrução condicional é exatamente como uma condição SELECT, como WHERE id=15 ou WHERE gender=”F”.

DELETE

DELETE é bastante auto-explicativa: você a utiliza para excluir permanentemente as informações do banco de dados. A sintaxe é:

DELETE datapoint FROM table WHERE condition;

A coisa mais importante a lembrar-se é da condição – se você não configurar uma, excluirá cada entrada nas colunas especificadas do banco de dados, sem uma confirmação ou uma segunda chance em muitos casos.

O que há em comum entre: DELETE x UPDATE

Você deve lembrar-se de utilizar uma condição toda vez que atualizar ou excluir. Se você não fizer isso, todas as linhas na tabela sofrerão a mesma alteração ou exclusão. Mesmo os programadores mais experientes esquecem a condição, para seu grande constrangimento profissional. Cuidado para não esquecer a condição.

CREATE

CREATE é utilizado para fazer uma tabela ou banco de dados completamente novo. Realmente criar um banco de dados implica um pouco mais que apenas atribuir um nome.

CREATE DATABASE db_name;

Todo o trabalho está em definir as tabelas e as colunas. Primeiro você declara o nome da tabela e depois deve detalhar os tipos específicos de colunas dessa tabela no que é chamado de “definição de criação”.

```
CREATE TABLE table {
    ->id_col INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ->col1 TEXT NULL INDEX,
```

```
        ->col2 DATE NOT NULL
};
```

Diferentes servidores de SQL têm tipos de dados e opções de definição ligeiramente diferentes, então a sintaxe de um não pode transferir exatamente para outro.

DROP

DROP pode ser utilizado para excluir completamente uma tabela ou banco de dados e todos os seus dados associados. Ele não é um comando mais sutil:

```
DROP TABLE table;
DROP DATABASE db_name
```

Obviamente é necessário muito cuidado com essa instrução.

ALTER

ALTER é a maneira de alterar uma estrutura da tabela. Você simplesmente indica que tabela está alterando e redefine suas especificações. Novamente, eis os produtos SQL diferem na sutileza.

```
ALTER TABLE table RENAME AS new_table;
ALTER TABLE new_table ADD COLUMN col3 VARCHAR(50) ;
ALTER TABLE new_table DROP COLUMN col2;
```

12. Bancos de dados compatíveis com o PHP:

Os bancos de dados atualmente suportados pelo PHP são: ***Adabas D, dBase, mSQL, InterBase, SyBase, Empress, MySQL, Velocis, FilePro, Oracle, dbm, Informix, PostgreSQL***.

13. Biblioteca de Funções

Neste capítulo não estão detalhadas todas as funções do PHP, mas grande parte das funções mais utilizadas estão detalhadas neste capítulo.

Bibliotecas requeridas

Para que se possa utilizar essas funções é preciso ter instalado, no servidor, as seguintes bibliotecas:

Módulo do PHP	Onde encontrar
LDAP	ftp://ftp.openldap.org/pub/openldap/ ftp://terminator.rs.itd.umich.edu/ldap/ ftp://ftp.critical-angle.com/pub/cai/slapd/
Berkley DB2	http://www.sleepycat.com/
SNMP	http://www.ece.ucdavis.edu/ucd-snmp/
GD	http://www.boutell.com/gd/#buildgd
mSQL	http://www.hughes.com.au/
MySQL	http://www.mysql.com/
IMAP	ftp://ftp.cac.washington.edu/imap/
FreeType (libttf)	http://www.freetype.org
Zlib	http://www.cdrom.com/pub/infozip/zlib
Expat XML parser	http://www.jclark.com/xml/expat.html
PDFlib	http://www.ifconnection.de/~tm/
mcrypt	ftp://argeas.cs-net.gr/pub/unix/mcrypt/
mhash	http://sasweb.de/mhash/
t1lib	http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/rmz/t1lib.html
Dmalloc	http://www.dmalloc.com/
Aspell	http://metalab.unc.edu/kevina/aspell/

Array

Array

Retorna um array de parâmetros. Índices podem ser atribuídos aos parâmetros através do operador.

Array array(...);

Exemplo:

$arr = array(“palavras” => array(“1” => “index”,”2” => “main“,”3” => “default”),”tradução” => array(“índice”,”principal”,”padrão”));

array_keys [PHP4]

Retorna os índices de um array.

Array array_keys(array matriz);

Array_merge [PHP4]

Retorna um array resultante da concatenação de dois ou mais arrays.
Array array_merge(array matriz1, array matriz2, [...]);

Array_pop[PHP4]

Retorna e remove o ultimo elemento de um array.
Mixed array_pop(array matriz1);

Array_push[PHP4]

Acrescente elementos no final de um array. Retorna o numero de elementos de uma array.
Int array_push(array matriz1, mixed elem1,[...]);

Array_shift[PHP4]

Retorna e remove o primeiro elemento de um array
Mixed array_shift(array array);

Array_slice[PHP4]

Retorna uma seqüência de elementos de um array.
Array array_slice(array matriz, int inicio, int [tamanho]) ;

Argumento	Descrição
Matriz	o array de onde serão copiados os elementos
Início	Posição inicial no array
tamanho	Número de elementos a serem retornados

Array_splice[PHP4]

Remove elementos de um array e, opcionalmente, substitui os elementos removidos por outros.
Array array_splice(array matriz, int início, int [tamanho], array [substituição]);

Argumento	Descrição
Matriz	o array de onde serão copiados os elementos
Início	Posição inicial no array entrada
Tamanho	Número de elementos a serem removidos
Substituição	Arrays dos elementos que substituirão os elementos removidos

Array_unshift[PHP4]

Acrescenta elementos no início de um array. Retorna o número de elementos de uma array.
Int array_unshift(array matriz, mixed var, [...]);

Array_values[PHP4]

Retorna os valores de um array
Array array_values(array matriz);

Array_walk

Aplica uma função para cada elemento de um array.
Int array_walk(array matriz, string nomefunc, mixed dadosextras);
Function nomefunc($item, $índice, $dadosextras)
Exemplo:

```
$matriz = array("1"=>"monitor","2"=>"Placa de vídeo", "3"=> "mouse");
//esta é uma das funções chamadas pelo array_walk
function imprime_item($item, $índice,$dados){
echo "$índice = $item($dados)";
}
//esta é uma das funções chamadas pelo array_walk
function atualizar_item(&$item, $índice, $dados){
$item = "$item ($dados)";
}
array_walk($matriz,'imprime_item','impresso');
array_walk($matriz,'atualizar_item','atualizado');
array_walk($matriz,'imprime_item','atualizado');
```

Arsort

Ordena um array na ordem reversa, mantendo a associação dos índices com os elementos
Void arsort(array array);

Asort

Ordena um array na ordem, mantendo a associação dos índices com os elementos
Void asort(array array);

Compact[PHP4]

Cria um array de variáveis com seus valores. Os parâmetros podem ser nomes de variáveis ou arrays com os nomes de variáveis.

Array compact(string nomevar|string nomesvar,[...]) ;
Exemplo:
$depto = “Desenvolvimento”;
$funcionário = “João Fulaninho”;
$cargo = “Analista de Sistemas” ;
$atributos = array(“cargo”,”depto”);
$res = compact(“funcionário”,”atributos”);
O código acima irá criar o array (“funcionário”=>”João Fulaninho”,”cargo”=:”Analista de Sistemas”,”depto”=>”Desenvolvimento”).

Count

Retorna o numero de elementos em um array. Retorna 1 se a variável não for um array e 0 se a variável não estiver definida.
Int count(mixed matriz) ;

Current

Retorna o elemento corrente em um array
Mixed current(array matriz);

Each

Retorna o par “elemento e valor” atual de um array, avançando o cursor do array para o próximo item. É semelhante a um loop *for* que percorre os elementos de um array.
Array each(array matriz) ;

End

Aponta o indicador interno de um array para o último elemento.
End(array matriz);

Extract

Converte os pares “índice => valor” em variáveis “nome => valor”. É o inverso da função compact().
Void extract(array matriz, int [tipo_extracação], string [prefixo]) ;

Tipo de extração	Descrição
EXTR_OVERWRITE	Sobrescreva as variáveis existentes se houver colisão
EXTR_SKIP	Não sobrescreva as variáveis se houver colisão
EXTR_PREFIX_SAME	Se houver colisão com as variáveis existentes, coloque o prefixo no nome da

	variável
EXTR_PREFIX_ALL	Coloque o prefixo no nome de todas as variáveis.

In_array[PHP4]

Retorna true se *texto* existir no array matriz
Bool in_array(mixed texto, array matriz);

Key

Retorna o elemento índice da posição corrente de um array
Mixed key(array matriz);

Ksort

Ordena um array pelos índices, mantendo a associação dos índices com os elementos.
Int ksort(array matriz) ;

List

Atribui uma lista de variáveis em apenas uma operação
Void list(...);

Next

Retorna o próximo elemento de um array, ou false se não houver mais elementos.
Mixed next(array matriz);

Pos

Retorna o elemento corrente de um array
Mixed pos(array matriz);

Prev

Retorna o elemento anterior ao elemento corrente de um array ou falso se não houver mais elementos.
Mixed prev(array matriz);

Range

Retorna um array contendo uma seqüência de números inteiros no intervalo especificado.
Array range(int lim_inferior, int lim_superior) ;

Reset

Aponta o indicador interno de um array para seu primeiro elemento.
Mixed reset(array matriz);

Rsort

Ordena um array na ordem inversa
Void rsort(array matriz);

Shuffle

Embaralha os elementos de um array
Void shuffle(array matriz);

Sizeof

Retorna o número de elementos de um array.
Int sizeof(array matriz);

Sort

Ordena um array
Void sort(array matriz);

Uasort

Ordena um array utilizando uma função definida pelo programador, mantendo a associação dos índices. A função de comparação deverá retornar um inteiro menor que zero, igual a zero, ou maior que zero, se o primeiro argumento for respectivamente menor que, igual a, ou maior que o segundo argumento.
Void uasort(array matriz, function func_compara) ;

Uksort

Ordena um array pelos índices, utilizando uma função definida pelo programador. A função de comparação deverá retornar um inteiro menor que zero, igual a zero, ou maior que zero, se o primeiro argumento for respectivamente menor que, igual a, ou maior que o segundo argumento.
Void uksort(array matriz, function func_compara);

Usort

Ordena um array pelos valores, utilizando uma função definida pelo programador. A função de comparação deverá retornar um inteiro menor que zero, igual a zero, ou maior que zero, se o primeiro argumento for respectivamente menor que, igual a, ou maior que o segundo argumento.
Void usort(array matriz, function func_compara);

Matemática para números inteiros

Bcpow

Eleva o primeiro número à potência do segundo número. O parâmetro adicional escala significa o número de dígitos após o ponto decimal.
String bcpow(string x, string y, int [escala]);

Bcscale

Muda o valor padrão do parâmetro escala para todas as funções matemáticas subseqüentes que não tenham um valor definido para escala.
String bcscale(int scala)

Datas

Checkdate

Retorna true se uma data for válida; caso contrário, retorna false
Int checkdate(int mês, int dia, int ano);

Date

Retorna uma data no formato especificado
String date(string formato,int[datahora]);

Argumento	Descrição
Data	Data/hora a ser formatada. Se não for especificada utilizará a data/hora corrente
formato	String com caracteres de formatação de datas

Caractere	Significado
a	“am” ou “pm”
A	“AM” ou “PM”
d	Dia em formato numérico. De “01” a “31”.
D	Dia da semana, textual, 3 letras. Ex.: “Fri”
F	Mês, textual, formato longo. Ex.: ”January”.

h	Hora no formato de 12 horas. De “01” a “12”.
H	Hora no formato de 24 horas. De “00” a “23”.
g	Hora no formato de 12 horas, sem os zeros à esquerda. De “1” a “12”.
G	Hora no formato de 24 horas, sem os zeros à esquerda. De “0” a “23”.
i	Minutos. De “00” a “59”.
j	Dia do mês sem os zeros à esquerda. De “1” a “31”.
l	Dia da semana, textual, longo. Ex.: “Friday”.
L	Booleano que indica se o ano é bissexto.”0” ou “1”.
m	Mês. De “01” a “12”.
n	Mês sem zeros à esquerda. De “1” a “12”.
M	Mês, textual, 3 letras. Ex.: “Jan”.
s	Segundos. De “00” a “59”.
S	Sufixo ordinal em inglês, textual, 2 caracteres. Ex.: “th“,”nd”.
t	Número de dias do mês determinado.Ex.:”28” a “31”.
U	Número de segundos desde o “epoch”.
w	Dia da semana, numérico. De “0”(domingo) a “6” (sábado) .
Y	Ano com 4 dígitos. Ex.: “2002”.
y	Ano com 2 dígitos. Ex.: “02”.
z	Dia do ano. De “0” a “365”.
Z	Fuso horário em segundos. De “-43200” a “43200”.

Getdate

Retorna um array associativo contendo todas as informações de uma data/hora específica.
Array getdate(int data/hora);

Índices da matriz retornada.

“seconds”	Segundos
“minutes”	Minutos
“hours”	Horas
“mday”	Dia do mês
“wday”	Dia da semana no formato decimal
“mon”	Mês no formato decimal
“year”	Ano no formato decimal
“yday”	Dia do ano no formato decimal
“weekday”	Dia da semana no formato decimal
“month”	Mês no formato texto

Gettimeofday

Retorna um array associativo contendo as informações da hora corrente
Array gettimeofday(void);

Os índices da matriz são:

“sec”	Segundos
“usec”	Microsegundos
“minuteswest”	Minutos a oeste de Greenwich
“dsttime”	Tipo da correção dst

Gmdate

Retorna uma data/hora GMT no formato UNIX. É idêntica à função mktime(), exceto que a hora retornada é a hora de Greenwich.
String gmdate(string formato, int datahora);

Gmmktime

Retorna a data/hora GMT no formato UNIX. É idêntica à função mktime(), exceto que a data passada no parâmetro representa uma data GMT.
Int gmmktime(int hora, int minuto, int segundo, int mês, int dia, int ano, int [dsf]);

Gmstrftime

Retorna uma data/hora GMT/CUT no formato especificado. É semelhante à função strftime(), exceto que a hora retornada é a hora de Greenwich.
String gmstrftime(string formato, int datahora);

Microtime

Retorna uma string contendo a data/hora atual no formato “msec sec” do UNIX, que significa o número de milisegundos e depois os segundos desde o UNIX Epoch(01/01/1970 00:00:00 GMT).
String microtime(void);

Mktime

Retorna uma data/hora no formato UNIX
Int mktime(int hora, int minuto, int segundo, int mês, ine dia, int ano, int [dsf]);

Strftime

Retorna uma data no formato especificado. Os nomes por extenso dos meses e dos dias da semana dependem da configuração feita com a função setlocale().
String strftime(string formato, int datahora);

Argumento	Descrição
Data	Data/hora a ser formatada. Se não especificada, utilizará a data/hora corrente

Formato String com caracteres de formatação de datas

Caractere	Significado
%a	Dia da semana abreviado
%A	Dia da semana completo
%b	Nome abreviado do mês
%B	Nome completo do mês
%c	Representação preferida de data e hora
%d	Dia do mês no formato decimal (00-31)
%H	Hora no formato decimal de 24 horas (00-23)
%l	Hora no formato decimal de 12 horas (00-12)
%j	Dia do ano no formato decimal (001-366)
%m	Mês em formato decimal (1-12)
%M	Minutos no formato decimal
%p	'am' ou 'pm', dependendo da hora especificada
%S	Segundos no formato decimal
%U	Número da semana do ano atual no formato decimal, começando no primeiro domingo como o primeiro dia da primeira semana
%W	Número da semana do ano atual no formato decimal, começando na primeira segunda-feira como o primeiro dia da primeira semana
%w	Dia da semana no formato decimal, sendo o domingo igual a 0.
%x	Representação preferida da data, sem a hora
%X	Representação preferida da hora, sem a data
%y	Ano no formato decimal, sem o século (00-99)
%Y	Ano no formato decimal, com o século.
%Z	Zona de fuso horário, ou nome, ou abreviação.
%%	Uma caractere literal "%".

Time

Retorna a data/hora no formato UNIX, que corresponde ao número de segundos desde o Unix Epoch(01\01\1970 00:00:00 GMT)
Int time(void);

Diretório

Chdir

Altera o diretório correspondente do PHP para o diretório especificado. Retorna true se tiver sucesso; caso contrário retorna false
Int chdir(string diretório);

Classe dir

É uma classe para leitura de diretórios. O diretório determinado é aberto. Esta classe possui duas propriedades que estarão disponíveis após a abertura do diretório: a propriedade handle, que pode ser usada com outras funções como readdir(), rewinddir() e closedir() e a propriedade path, que contém o caminho do diretório que foi aberto. Três métodos estão disponíveis: read, rewind e close.
New dir(string diretório) ;

Closedir

Encerra a associação de um identificador(handle) com um diretório.
Void closedir(int handle_dir) ;

Opendir

Retorna um handle de diretório para ser usado nas funções closedir(), readdir() e rewinddir().
Int opendir(string diretório);

Readdir

Retorna o próximo nome de arquivo do diretório, na ordem em que estiverem armazenadas.
String readdir(int handle_dir);

Rewinddir

Retorna o ponteiro à posição inicial do diretório
Void rewinddir(int handle_dir);

Execução de Programas

Escapeshellcmd

Retira quaisquer caracteres de um string que poderiam ser utilizados para enganar um comando Shell para executar comandos arbitrários. Esta função normalmente é usada para verificar os dados fornecidos pelo usuário antes de serem passados para as funções exec() ou system().
String escapeshellcmd(string comando);

Exec

Executa um comando externo e mostra a última linha do resultado do comando.
String exec(string comando, string [array], int [variável_ref]) ;

Argumento	Descrição
Comando	Comando externo a ser executado
Array	Array contendo as linhas do resultado
Variável_ref	Variável que conterá o código de retorno do comando executado

Passthru

Executa um comando externo e mostra todos os resultados.
String passthru(string comando, int [variável_ref]);

Argumento	Descrição
Comando	Comando externo a ser executado
Variável_ref	Variável que conterá o código de retorno do comando executado

System

Executa um comando externo e mostra os resultados.
String system(string comando, int [variável_ref]);

Argumento	Descrição
Comando	Comando externo a ser executado
Variável_ref	Variável que conterá o código de retorno do comando executado

Sistema de arquivos do servidor

Basename

Retorna o nome do arquivo em um path. No windows, ambos os caracteres, / e \, são usados como separadores de diretórios. Em outros ambientes é utilizado somente o caractere /.
String basename(string path);

Chgrp

Muda o grupo ao qual um arquivo pertence em ambientes UNIX. Em ambientes windows não faz nada, retornando sempre true.
Ine chgrp(string nome_do_arquivo, mixed grupo) ;

Chmod

Altera as permissões de um arquivo em ambientes UNIX. Retorna true em caso de sucesso; caso contrário, retorna false.
Em ambientes windows esta função não faz nada, sempre retornando true
Int chmod(string nome_do_arquivo, int modo) ;

Argumento	Descrição

Nome_arquivo	Nome do arquivo
Modo	O valor passado como parâmetro deve ser do tipo octal. Ex.: chmod("/algumdiretorio/algumarquivo",0755);

Chown

Altera o proprietário de um arquivo em ambientes UNIX. Somente o root pode alterar o proprietário de um arquivo. Retorna true em caso de sucesso; caso contrário, retorna false. No windows esta função na faz nada sempre retornando true.
Int chown(string nome_arquivo, mixed proprietário);

Clearstatcache

Apaga o cache de status dos arquivos utilizados em tempo de execução no PHP
Void clearstatcache(void);

Copy

Copia um arquivo. Retorna true se tiver sucesso; caso contrário, retorna false.
Int copy(string nome_arquivo, string arquivo_destino);

Delete

Apaga um arquivo do servidor
Void delete(string nome_arquivo) ;

Dirname

Retorna o nome de um diretório em um path. Veja a função basename().
String dirname(string path);

Diskfreespace

Retorna o número de bytes disponíveis em um disco.
Int diskfreespace(string nome_arquivo);

Fclose

Fecha um arquivo. Retorna true se tiver sucesso; caso contrário, retorna false.
Int fclose(int fp);

Feof

Retorna true caso o ponteiro de arquivo esteja no fim do arquivo ou ocorra um erro; caso contrário, retorna false.
Int feof(fp);

Fgetc

Retorna um caractere apontado pelo ponteiro do arquivo. Retorna false se for EOF.
String fgetc(int fp);

Fgetcsv

Lê uma linha do arquivo a partir do ponteiro e coloca em um array os campos lidos. Linhas em branco serão convertidas em arrays compreendendo um único campo nulo.
Array fgetcsv(int fp, int tamanho, string [delimitador]);

Fgets

Retorna uma string do arquivo a partir do ponteiro de arquivo até que (tamanho - 1) bytes sejam lidos do arquivo. A leitura termina quando encontra o fim de uma linha ou quando o arquivo termina.
String fgets(int fp, int tamanho);

Fgetss

Lê strings do arquivo como a função fgets(), tirando todas as tags HTML do texto lido.
String fgetss(int fp, int tamanho) ;

File

Lê um arquivo inteiro e coloca seu conteúdo em um array. Cada elemento do array corresponderá a cada linha do arquivo lido, sem tirar qualquer informação.
Array file(string nome_arquivo);

File_exists

Retorna true se um determinado arquivo existe; caso contrário, retorna false. Os resultados desta função são armazenados e reutilizados. Para atualizar esta informação utilize a função clearstatcache().
Int file_exists(string nome_arquivo);

Fileatime

Retorna a data/hora do último acesso ao arquivo, ou false se ocorrer um erro. Para atualizar esta informação utilize a função clearstatcache().
Int fileatime(string nome_arquivo);

Filectime

Retorna a data/hora da última mudança do inode de um arquivo, ou false se ocorrer um erro. Para atualizar esta informação utilize a função clearstatcache().
Int filectime(string nome_arquivo);

Filegroup

Retorna o nome do grupo ao qual pertence o arquivo, ou false em caso de erro. Os resultados desta função são armazenadas e reutilizadas. Para utilizar esta informação utilize a função clearstatcache().
Int filegroup(string nome_arquivo);

Fileinode

Retorna o número do inode de um arquivo, ou false em caso de erro. Os resultados desta função são armazenadas e reutilizadas. Para utilizar esta informação utilize a função clearstatcache().
Int fileinode(string nome_arquivo);

Filemtime

Retorna a data/hora da última alteração de um arquivo, ou false se ocorrer um erro. Para atualizar esta informação utilize a função clearstatcache().
Int filemtime(string nome_arquivo);

Fileowner

Retorna o número de ID do dono de um arquivo, ou false em caso de erro. Os resultados desta função são armazenadas e reutilizadas. Para utilizar esta informação utilize a função clearstatcache().
Int fileowner(string nome_arquivo);

Fileperms

Retorna as permissões de um arquivo, ou false em caso de erro. Os resultados desta função são armazenadas e reutilizadas. Para utilizar esta informação utilize a função clearstatcache().
Int fileperms(string nome_arquivo);

Filesize

Retorna o tamanho de um arquivo, ou false em caso de erro. Os resultados desta função são armazenadas e reutilizadas. Para utilizar esta informação utilize a função clearstatcache().
Int filesize(string nome_arquivo);

Filetype

Retorna o tipo de um arquivo. Os valores possíveis são fifo, char, dir, block, link, file e unknown. Retorna false em caso de erro. Os resultados desta função são armazenadas e reutilizadas. Para utilizar esta informação utilize a função clearstatcache().
String filetype(string nome_arquivo);

Flock

Altera as opções de compartilhamento de um arquivo aberto, fazendo com que todos os programas que precisem usar esse arquivo usem as mesmas opções de compartilhamento. Retorna true se tiver sucesso; caso contrário, retorna false.
Bool flock(int fp, int operacao);

Argumento	Descrição
Operação	Tipo de operação
1	Modo compartilhado para leitura
2	Modo exclusivo para escrita
3	Modo livre (compartilhado ou exclusivo)
4	Adicione 4 para não bloquear o arquivo enquanto estiver alterando o modo

Fopen

Abre um arquivo ou uma URL. A função retornará false se ocorrer erro.
Int fopen(string nome_arquivo, string modo);

Argumento	Descrição
Nome_arquivo	Nome do arquivo. A ação a ser tomada dependerá de como o nome é iniciado. Veja a baixo as opções.
“http://”	Abre um conexão HTTP 1.0 com o servidor especificado e um ponteiro de arquivo é retornado ao início do texto da resposta
“ftp://”	Abre um conexão FTP com o servidor especificado e um ponteiro de arquivo é retornado. O servidor deve suportar o modo passivo de FTP. É possível abrir os arquivos para leitura ou para escrita, mas não ambos ao mesmo tempo
outro	Se o nome começar com qualquer outra coisa, o arquivo será aberto no sistema de arquivos e um ponteiro de arquivo será retornado.
Modo	Modo de abertura. Adicionalmente pode-se colocar a letra “b” no parâmetro modo, informando que a ser processado é um arquivo binário
“r”	Somente leitura, a partir do inicio do arquivo.
“r+”	Leitura e escrita, a partir do inicio do arquivo.
“w”	Somente escrita. A partir do inicio do arquivo e apagando todo o conteúdo

	do arquivo. Se o arquivo não existir, a função tentará criá-lo.
“w+”	Para leitura e escrita. A partir do inicio do arquivo e apagando todo o conteúdo do arquivo. Se o arquivo não existir, a função tentará criá-lo.
“a”	Somente escrita. A partir do inicio do arquivo. Se o arquivo não existir, a função tentará criá-lo.
“a+”	Para leitura e escrita. A partir do inicio do arquivo. Se o arquivo não existir, a função tentará criá-lo.

Fpassthru

Lê o conteúdo do arquivo, do ponteiro ate o fim do arquivo, mostrando os dados e fechando o arquivo logo em seguida.
Int passthru(int fp);

Fputs

Escreve o conteúdo de uma string em um arquivo
Int fputs(int fp, string str, int [tamanho]);

Fread

Lê bytes de um arquivo
String fread(int fp, int [tamanho]);

Fseek

Muda a posição do ponteiro de arquivo. Só pode ser usada em arquivos do sistema de arquivos local. Retorna 0 se tiver sucesso; caso contrário, retorna –1.
Int fseek(fp, int offset);

Ftell

Retorna a posição do ponteiro do arquivo
Int ftell(int fp);

Fwrite

Escreve um número especifico de bytes de uma string em um arquivo
Int fwrite(int fp, string str, int [numbytes]);

Is_dir

Retorna true se o nome do arquivo especificado existir e for um diretório. Os resultados desta função são armazenadas e reutilizadas. Para atualizar esta informação utilize clearstatcache().
Bool is_dir(string nome_arquivo);

Is_executable

Retorna true se o nome do arquivo especificado existir e for um arquivo executável. Os resultados desta função são armazenadas e reutilizadas. Para atualizar esta informação utilize clearstatcache().
Bool is_executable(string nome_arquivo);

Is_file

Retorna true se o nome do arquivo especificado existir e for um arquivo. Os resultados desta função são armazenadas e reutilizadas. Para atualizar esta informação utilize clearstatcache().
Bool is_file(string nome_arquivo);

Is_link

Retorna true se o nome do arquivo especificado existir e for um link simbólico. Os resultados desta função são armazenadas e reutilizadas. Para atualizar esta informação utilize clearstatcache().
Bool is_link(string nome_arquivo);

Is_readable

Retorna true se o nome do arquivo especificado existir e puder ser lido. Os resultados desta função são armazenadas e reutilizadas. Para atualizar esta informação utilize clearstatcache().
Bool is_readable(string nome_arquivo);

Is writeable

Retorna true se o nome do arquivo especificado existir e puder ser escrito. Os resultados desta função são armazenadas e reutilizadas. Para atualizar esta informação utilize clearstatcache().
Bool is_writeable(string nome_arquivo);

Link

Cria um hard link
Int link(string target, string link);

Linkinfo

Retorna informações sobre um link
Int linkinfo(string path);

Mkdir

Cria um diretório com o modo especificado em formato octal
Retorna true se tiver sucesso; caso contrário, retorna false.
Int mkdir(string pathname, modo);

Pclose

Fecha um ponteiro de arquivo aberto por popen(), retornando o status do processo executado.
Int pclose(int fp);

Popen

Abre um ponteiro para um processo de arquivo
Int popen(string comando, string modo);

Readfile

Lê um arquivo e o envia para a saída padrão do sistema; retorna o número de bytes lidos
Int readfile(string nome_arquivo);

Readlink

Retorna o alvo(target) de um link simbólico
String readlink(string path);

Rename

Altera o nome de um arquivo. Retorna true se tiver sucesso; caso contrário false
Int rename(string nome_anterior, string nome_novo);

Rewind

Muda a posição do ponteiro para o inicio do arquivo
Int rewind(int fp);

Rmdir

Apaga um diretório vazio
Int rmdir(string nome_diretorio);

Set_file_buffer

Define o tamanho do buffer de escrita de arquivos (default = 8192 bytes). Se definido como 0, então as operações de escrita não serão colocadas em buffer.

Stat

Retorna em um array as informações sobre um arquivo
Array stat(string nome_arquivo) ;

Elemento	Conteúdo
1	Device
2	Inode
3	Modo de proteção inode
4	Número de links
5	Identificação do usuário proprietário do arquivo
6	Identificação do grupo proprietário do arquivo
7	Tipo de dispositivo Inode(não disponível no windows)
8	Tamanho(em bytes)
9	Data/hora do último acesso
10	Data/hora da última alteração no arquivo
11	Data/hora da última alteração de estado do arquivo
12	Tamanho de bloco para filesystem I/O (não disponível no windows)
13	Número de blocos alocados

Symlink

Cria um link simbólico entre o alvo existente e o nome especificado
Int symlink(string target, string link);

Tempnam

Retorna um nome de arquivo temporário único no diretório especificado com um prefixo determinado pelo parâmetro prefixo
String tempnam(string diretório, string prefixo);

Touch

Altera a data/hora de modificação de um arquivo. Se o arquivo não existir, será criado. Retorna true se tiver sucesso; caso contrário, retorna false.
Int touch(string nome_arquivo, int time);

Umask

Altera o umask atual. Quando o PHP é um módulo do servidor, o umask é restaurado no fim de cada request.
Int umask(int mask) ;

Unlink

Elimina um nome de arquivo(link)
Int unlink(string nome-arquivo);

Opções e informações do PHP

Error_log

Envia uma mensagem de erro para o registro de erros do servidor web, para uma porta TCP ou para um arquivo
Int error_log(string mensagem, int tipo_mensagem, string [destino], string [headersextras]);

Argumento	Descrição
Mensagem	Mensagem de erro
Tipo_mensagem	Tipo da mensagem
0	A mensagem é enviada para o log do php
1	A mensagem é enviada por email para o endereço indicado no argumento destino
2	A mensagem é enviada através de uma conexão remota de debugging do PHP.
3	A mensagem é adicionada ao arquivo indicado no argumento destino.
Destino	Destino da mensagem. Depende do tipo da mensagem
Headersextras	Argumento opcional utilizado quando o argumento tipo_mensagem for 1.

Error_reporting

Define quais error devem ser reportados pelo PHP
Int error_reporting(int [nível]);

Argumento	Descrição
Nível	Valor obtido com a soma dos códigos dos erros que devem ser reportados
1	E_ERROR – Erros que o PHP não pode ignorar, como falhas de alocação de memória, etc.
2	E_WARNING – Error que o PHP suspeita poderem causar um erro mais grave
4	E_PARSE – Erro de processamento do programa
8	E_NOTICE – Avisos de que certa parte de um código pode causar um erro mais grave
16	E_CORE_ERROR – Erros graves com os E_ERROR, mas são gerados pela

base PHP.
32 E_CORE_WARNING – Possíveis erros no código, mas gerados pela base PHP

Extension_loaded

Retorna true se uma extensão foi carregada. Utilize a função phpinfo() para ver os nomes das extensões.
Bool extension_loaded(string nome_extensão);

Get_cfg_var

Retorna o valor de uma opção de configuração do PHP, ou false se ocorrer um erro
String get_cfg_var(string nome_variável) ;

Get_current_user

Retorna o nome do proprietário do script PHP corrente
String get_current_user(void);

Get_magic_quotes_gpc

Retorna o status da opção magic_quotes_gpc do arquivo de configuração php.ini (0 para off, 1 para on).
Long get_magic_quotes_gpc(void) ;

Get_magic_quotes_runtime

Retorna o status da opção magic_quotes_runtime do arquivo de configuração php.ini (0 para off, 1 para on).
Long get_magic_quotes_runtime(void) ;

Getenv

Retorna o valor de uma variável de ambiente, ou false se ocorrer um erro. Para ver a lista de variáveis de ambiente utilize a função phpinfo().
String getenv(string nome_variável);

Getlastmod

Retorna a data/hora da última modificação da página corrente ou false se ocorrer um erro
Int getlastmod(void);

Getmyinode

Retorna o inode do script corrente, ou false se ocorrer um erro.
Int getmyinode(void);

Getmypid

Retorna o ID de processo corrente do PHP, ou false se ocorrer um erro
Int getmypid(void);

Getmyuid

Retorna o ID do usuário do script PHP corrente, ou false ocorrer um erro
int getmyuid(void) ;

Getrusage

Retorna informações da utilização dos recursos atuais em um array
Array getrusage(int [who]);

Phpinfo

Retorna diversas informações sobre o PHP
Int phpinfo(void);

Phpversion

Retorna a versão do PHP instalada.
String phpversion(void);

Putenv

Atribui um valor a uma variável de ambiente
Void putenv(string setting);

Set_magic_quotes_runtime

Altera a configuração atual da opção magic_quotes_runtime do arquivo de configuração php.ini (0 para off, 1 para on)
Long set_magic_quotes_runtime(int nova_conf);

Set_time_limit

Limita o tempo Maximo de execução do script. Se definido como 0(zero) o script será executado indefinidamente.
Void set_time_limit(int num_segundos);

Matemática

Abs

Retorna o valor absoluto de um número.
Mixed abs(mixed número);

Acos

Retorna o arco-coseno de um ângulo em radianos
Float acos(float arg);

Asin

Retorna o arco-seno de um ângulo em radianos
Float asin(float arg);

Atan

Retorna o arco-tangente de um ângulo em radianos
Float atan(float arg);

Atan2

Retorna o arco-tangente de duas variáveis
Float atan2(float y, float x);

Base_convert

Retorna uma string com um número convertido para outra base numérica.
String base_convert(string número, int base_ant, int nova_base);

Bindec

Retorna um número convertido de binário para decimal
Int bindec(string string_binário);

Argumento	Descrição
String_binário	String contendo a representação binária de um número

Ceil

Retorna o próximo número inteiro maior ou igual ao número especificado
Int ceil(float número);

Cos

Retorna o co-seno de um ângulo em radianos
Float cos(float arg);

Decbin

Retorna uma string contendo a representação binária de um número
String decbin(int número);

Dechex

Retorna uma string contendo a representação hexadecimal de um número
String dechex(int número);

Decoct

Retorna uma string contendo a representação octal de um número
String decoct(int número);

Exp

Retorna a constante e elevada à potência especificada
Float exp(float arg);

Floor

Retorna o próximo número inteiro menor ou igual ao número especificado
Int floor(float número);

Getrandmax

Retorna o valor máximo que pode ser gerado pela função rand()
Int getrandmax(void);

Hexdec

Retorna um número convertido de hexadecimal para decimal.
Int hexdec(string string_hexa);

Log

Retorna o logaritmo natural de um número
Float log(float arg) ;

Log10

Retorna o logaritmo base 10 de um número
Float log10(float arg) ;

Max

Retorna o maior dentre os especificados
Mixed max(mixed arg1, mixed arg2, mixed argn);

Min

Retorna o menor dentre os especificados
Mixed min(mixed arg1, mixed arg2, mixed argn);

Mt_rand

Gera um número aleatório mais confiável no intervalo especificado
Int mt_rand(int [limite_inf], int [limite_sup]);

Mt_srand

Altera a semente do gerador de números aleatórios para a função mt_rand()
Void mt_srand(int semente) ;

Mt_getrandmax

Retorna o maior valor possível que a função mt_rand() pode retornar
Int mt_getrandmax(void);

Number_format

Retorna a versão formatada de um número, colocando os separadores de milhares e o separador decimal
String number_format(float número, int decimais, string sep_dec, string sep_milhar);

Octdec

Retorna um numero convertido de octal para decimal
Int octdec(string octal_string);

Pi

Retorna o valor da constante pi
Double pi(void);

Pow

Retorna o resultado de uma base elevada a um expoente
Float pow(float base, float expoente);

Rand

Retorna um número aleatório dentro de um intervalo especificado
Int rand(int [limit_inf], int [limi_sup]);

Round

Retorna o valor de um número arredondado para o número inteiro mais próximo. Se o número estiver exatamente entre dois números inteiros, o resultado será sempre o número inteiro par
Double rand(double número);

Sin

Retorna o seno de um ângulo em radianos
Float sin(float arg);

Sqrt

Retorna a raiz quadrada
Float sqrt(float arg);

Srand

Altera a semente do gerador de números aleatórios para a função
Void srand(int semente);

Tan

Retorna a tangente de um ângulo em radianos
Float tan(float arg);

Criptografia

Mcrypt_cbc

Criptografa/descriptografa dados no modo CBC
Int mcrypt_cbc(int cifra, string chave, string dados, int modo, string vetor_ini);

Mcrypt_cfb

Criptografa/descriptografa dados no modo CFB
Int mcrypt_cfb(int cifra, string chave, string dados, int modo, string vetor_ini);

Mcrypt_create_iv

Cria um vetor de inicialização (vetor_ini) de uma origem aleatória, que é utilizado como argumento nas funções de criptografia.
String mcrypt_create_iv(int tamanhoiv, int origemiv);

Mcrypt_ecb

Criptografa/descriptografa dados no modo ECB
Int mcrypt_ecb(int cifra, string chave, string dados, int modo, string vetor_ini);

Mcrypt_get_cipher_name

Retorna o nome da cifra especificada, ou false se a cifra não existir
String mcrypt_get_cipher_name(int cifra);

Mcrypt_get_block_size

Retorna o tamanho (em bytes) do bloco de uma cifra
Int mcrypt_get_block_size(int cifra);

Mcrypt_get_key_size

Retorna o tamanho (em bytes) da chave de uma cifra
Int mcrypt_get_key_size(int cifra);

Mcrypt_ofb

Criptografa/descriptografa dados no modo OFB
Int mcrypt_ofb(int cifra, string chave, string dados, int modo, string vetor_ini);

Funções diversas

Connection_aborted

Retorna true se um cliente desconectou
Int connection_aborted(void);

Connection_status

Retorna o status de uma conexão
Int connection_status(void);

Connection_timeout

Retorna true se o tempo de execução do script foi excedido

Int connection_timeout(void);

Dl

Carrega, em tempo de execução, uma extensão do PHP definida em library. A diretiva de configuração extension_*dir* do arquivo de configuração php.ini define em qual diretório o PHP deve procurar as extensões carregadas dinamicamente
Int dl(string library) ;

Eval

Processa uma string como código PHP
Void eval(string string_código);

Die

Imprime uma mensagem e termina o script PHP
Void die(string mensagem);

Exit

Termina o script PHP atual
Void exit(void);

Function_exists

Retorna true se uma determinada função foi definida; caso contrário, retorna false
Int function_exists(string nome_função);

Ignore_user_abort

Configura o processador do PHP para não cancelar o script caso o cliente desconecte antes da finalização do mesmo
Int ignore_user_abort(int [config]);

Iptcparse

Retorna uma lista das tags de um bloco binário IPTC (http://www.xe.net/iptc/)
Array iptcparse(string bloco_iptc);

Leak

Limpa a memória desperdiçada
Void leak(int numbytes);

Mail

Envia um email para um ou mais destinatários
Bool mail(string dest, string assunto, string mensagem, string [headers_adic]);

Argumento	Descrição
Dest	Endereço de email dos destinatários(separados por “,”)
Assunto	Assunto do email
Mensagem	Conteúdo do email
Headers_adic	Especifica headers adicionais que devem ser inseridas no fim do header padrão. Múltiplos headers podem ser especificados e devem ser separados por newline(“\n”)

Exemplo:

```
$destino = “NomeDestino <endereço@dominio.com.br>”;
$remetente = “NomeRemetente <remetente@dominio.com.br>”;
$assunto = “assunto do email”;
mail($destino,”$assunto”, $mensagem, “From: $remetente \n” );
```

Pack

compacta dados em uma string binária
string pack(string formato, mixed [args]...) ;

Register_shutdown_function

Registra uma função para execução ao término do script
Int register_shutdown_function(string nome_funcao);

Serialize

Gera uma apresentação armazenável de um valor
String serialize(mixed valor);

Sleep

Atrasa a execução por um tempo determinado (em segundos)
Void sleep(int num_segundos)

Unpack

Descompacta dados numa string binária
Array unpack(string formato, string data);

Unserialize

Gera um valor válido a partir de uma string em representação para armazenameto
Mixed unserialize(string str);

Uniquid

Gera um identificador único baseado na hora atual em microsegundos
Int uniquid (string prefixo);

Usleep

Atrasa a execução por um tempo determinado (em microsegundos)
Void usleep(int num_microsegundos);

Rede

Checkdnsrr

Verifica se um nome de servidor ou endereço de Ip existe nos registros de DNS
Int checkdnsrr(string host, string [tipo])

Closelog

Fecha uma conexão ao histórico do sistema (system logger)
Int closelog(void) ;

Debugger_on

Habilita o depurador interno do PHP
Int debugger_on(string endereço);

Debugger_off

Desabilita o depurador interno do PHP
Int debugger_off(void);

Fsockopen

Abre uma conexão com um servidor
Int fsockopen(string hostname, int port, int [errno], string [errstr], double [timeout]);

Gethostbyaddr

Retorna o nome do servidor de um endereço IP
String gethostbyaddr(string endereço_IP);

Gethostbyname

Retorna o endereço IP de um servidor
String gethostbyname(string hostname);

Openlog

Abre uma conexão ao histórico do sistema(system logger) para um programa
Int openlog(string ident, int opção, int facilidade);

Pfsockopen

Abre uma conexão persistente com um servidor
Int pfsockopen(string hostname, int port, int [errno], string [errstr], double [timeout]);

Set_socket_blocking

Define o modo de conexão com o servidor (bloqueado ou não bloqueado)
Int set_socket_blocking(int socket, int modo);

Syslog

Gera uma mensagem de histórico (log) do sistema
Int syslog(int prioridade, string mensagem);

Expressões regulares

Ereg

Expressão regular de busca/comparação.
Int ereg(string expr_regular, string texto, array [regs]);

Argumento	Descrição
Expr_regular	Expressão regular
Texto	Texto para busca
Regs	Opções diversas. Veja “man regex”

Ereg_replace

Expressão de busca/substituição. Retorna a string modificada
String ereg_replace(string expr_regular, string texto_subst, string texto) ;

Argumento	Descrição
Expr_regular	Expressão regular
Texto_subst	Texto para substituição
Texto	Texto a ser substituído

Eregi

Similar à função ereg(), exceto que não é sensível a letras maiúsculas e minúsculas
Int eregi(string expr_regular, string texto, array [regs]);

Eregi_replace

Similar à função ereg_replace(), exceto que não é sensível a letras maiúsculas e minúsculas
String eregi_replace(string expr_regular, string texto_subst, string texto) ;

Split

Expressão regular para dividir uma string em partes e colocar as partes em um array
Array split(string expr_regular, string texto, int [limite]);

Argumento	Descrição
Expr_regular	Expressão regular
Texto	String a ser dividida
Limite	Limitador (caractere que caracteriza os pontos de divisão na string)

Sql_regcase

Retorna uma expressão regular para busca não-sensível a letras maiúsculas e minúsculas
String sql_regcase(string string);

Tratamento de sessões

Session_decode[PHP4]

Decodifica os dados de uma sessão em uma string
Bool session_decode(string dados);

Session_destroy[PHP4]

Destrói os dados registrados associados à sessão atual
Bool session_destroy(void);

Session_encode[PHP4]

Codifica os dados de uma sessão
Bool session_encode(void) ;

Session_start[PHP4]

Inicializa os dados de uma sessão

Bool session_start(void);

Session_id[PHP4]

Retorna ou muda o identificador da sessão atual
String session_id(string [id]);

Session_is_registered[PHP4]

Descobre se uma variável foi registrada na sessão
Bool session_is_registered(string nome);

Session_module_name[PHP4]

Retorna ou muda o nome do módulo da sessão atual
String session_module_name(string [módulo]);

Session_name[PHP4]

Retorna ou muda o nome da sessão atual
String session_name(string [nome]);

Session_register[PHP4]

Registra uma variável com a sessão atual
Bool session_register(string nome);

Session_save_path[PHP4]

Retorna ou muda o path de gravação da sessão atual
String session_save_path(string [path]);

Session_unregister[PHP4]

Descarta uma variável da sessão atual
Bool session_unregister(string nome);

Strings

Addslashes

Coloca barras invertidas antes dos caracteres especiais: apóstrofo(‘), aspas(“), barra invertida (\) e NUL
String addslashes(string str);

Bin2hex

Converte dados binários em representação hexadecimal
String bin2hex(string str);

Chop

Remove espaços em branco em seqüência
String chop(string str);

Chr

Retorna um caractere especifico dado um código ASCII
String chr(int cód_ascii);

Chunk_split

Divide uma string em pedaços de tamanho “compr”, colocando a string “fim” no fim de cada pedaço e retornando tudo em uma string. O parâmetro “compr” tem valor padrão de 76 e o parâmetro “fim” tem valor padrão de “\r\n” caso esses não sejam especificados.
String chunck_split(string texto, int [compr], string [fim]);

Convert_cyr_string

Converte uma string de um conjunto de caracteres cirílicos em outro
String convert_cyr_string(string str, string de, string para);

Crypt

Retorna uma string criptografada através do modo DES.
String crypt(string str, string [salf]);

Echo

Imprime uma ou mais strings
Echo (string arg1, string argn....);

Explode

Retorna um array contendo as partes da string com valores separados por um separador
Array explode(string separador, string str) ;

Flush

Envia todo o buffer de impressão ao browser cliente
Void flush(void);

Get_meta_tags

Extrai todas as tags <meta> de um arquivo e retorna um array
Array get_meta_tags(string nome_arquivo, int [use_include_path]) ;

Htmlentities

Converte todos os caracteres aplicáveis em tags HTML
String htmlentities(string str);

Htmlspecialchars

Converte caracteres especiais em tags HTML
String htmlspecialchars(string str);

Caracter	Descrição
‘&’	‘&’
‘”’	‘"’
‘<’	‘<’
‘>’	‘>’

Implode

Retorna uma string contendo a representação string de todos os elementos de um array separados pelo argumento glue
String implode(string glue, array fatias);

Join

O mesmo que implode()
String join(string glue, array fatias);

Ltrim

Retorna uma string sem os espaços iniciais
String ltrim(string str);

Md5

Retorna o hash MD5 de uma string
String md5(string str);

Nl2br

Retorna uma string com tags <BR> inseridos após os caracteres de nova linha
String nl2br(string str);

Ord

Retorna o valor ASCII do primeiro caractere de uma string
Int ord(string str);

Parse_str

Converte uma string de parâmetros no formato das URLs vindas de formulários HTML em variáveis
Void parse_str(string str);
Exemplo:

```
$str = "form=firm&operation=insert&index=0";
parse_str($str);
echo $form;   //imprime "firm"
echo $operation;   //imprime "insert"
echo $index;   //imprime "0"
```

Print

imprime uma string
print(string arg);

Printf

Imprime uma string formatada
Int printf(string formato, mixed [args]...);

Quoted_printable_decode

Converte uma string imprimível em uma string de 8 bits
String quoted_printable_decode(string str);

Quotemeta

Coloca uma barra invertida antes dos caracteres meta (\\ + * ? [^] ($)) de uma string
String quotemeta(string str);

Rawurldeconde

Decodifica uma URL, retornando uma URL com os caracteres na forma literal
String rawurldecode(string str);

Rawurlenconde

Codifica uma URL, retornando uma URL com os caracteres não-alfanuméricos convertidos em códigos
String rawurlencode(string str);

Setlocale

Altera as informações de localidade
String setlocale(string categoria, string locale);

Similar_text

Calcula a similaridade entre duas strings
Int similar_text(string str1, string str2, double [porcentagem]);

Soundex

Calcula a chave soundex de uma string
String soundex(string str);

Sprintf

Retorna uma string formatada
String Sprintf(string formato, mixed [args]);

Strchr

O mesmo que strstr()
String strchr(string haystack, string needle);

Strcmp

Compara duas strings, retornando um valor: < 0 se str1 for menor que str2; > 0 se str1 for maior que str2, e 0 se elas forem iguais
Int strcmp(string str1, string str2);

Strcspn

Retorna o comprimento do segmento inicial de str1 que não contem nenhum dos caracteres de str2
Int strcspn(string str1, string str2);

Strip_tags

Retorna uma string sem as tags HTML e PHP
String strp_tags(string str);

Stripslashes

Apaga as barras invertidas de caracteres específicos
String stripslashes(string str);

Strlen

Retorna o comprimento de uma string
Int strlen(string str);

Strpos

Retorna a posição da primeira ocorrência de uma string(str2) dentro de outra (str1)
Int strpos(string str1, string str2, int [offset]);

Strrpos

Retorna a posição da última ocorrência de uma string(str2) dentro de outra(str1)
Int strrpos(string str1, char str2);

Strrchr

Retorna a parte do texto que vai do fim da string até a primeira ocorrência do caracter "car". Retorna false se não houver nenhuma ocorrência de car
String strrchr(string texto, string car);

Strrev

Inverte uma string
String strrev(string str);

Strspn

Retorna o comprimento do segmento inicial de str1 que consiste inteiramente de caracteres em str2
Int strspn(string str1, string str2);

Strstr

Retorna a parte de str1 que vai da primeira ocorrência de str2 em str1 até o fim de str1
String strstr(string str1, string str2);

Strtok

Separa uma string em partes divididas por arg2
String strtok(string arg1, string str2) ;

Strtolower

Transforma as letras em uma string para minúscula.
String strtolower(string str) ;

Strtoupper

Transforma as letras em uma string para maiúscula.
String strtoupper(string str) ;

Str_replace

Substitui todas as ocorrências de uma substring por outra
String str_replace(string str_ant, string novo_str, string str) ;

Argumento	Descrição
Str_ant	Substring a ser substituída
Novo_str	Substring que substituirá a anterior
Str	String original

Strtr

Traduz todas as ocorrências de cada caractere em *de* pelo caractere correspondente em *para* na string str
String strtr(string str, string de, string para);

Substr

Retorna parte de uma string
String substr(string.str, int inicio, int [tamanho]);

Argumento	Descrição
Inicio	posição inicial
Tamanho	Número de caracteres a serem retornados

Trim

Apaga os espaços em branco do inicio e fim de uma string.
String trim(string str);

Ucfirst

Transforma o primeiro caractere de uma string em maiúsculo
String ucfirst(string str) ;

Ucwords

Transforma o primeiro caractere de cada palavra de uma string em maiúsculo
String ucwords(string str);

Funções para variáveis

Doubleval

Retorna o valor em ponto flutuante de uma variável
Double doubleval(mixed var) ;

Empty

Retorna false se var estiver atribuída; caso contrário retorna true
Int empty(mixed var) ;

Gettype

Retorna o tipo de uma variável
String gettype(mixed var);

Intval

Retorna o valor em inteiros de uma variável, utilizando uma base especificada. O padrão da base é 10
Int intval(mixed var, int [base])

Is_array

Retorna true se a variável for do tipo array
Int is_array(mixed var);

Is_double

Retorna true se a variável for do tipo double
Int is_double(mixed var);

Is_float

Retorna true se a variável for do tipo float

Int is_float(mixed var);

Is_int

Retorna true se a variável for do tipo inteiro
Int is_int(mixed var);

Is_integer

O mesmo que is_int()
Int is_integer(mixed var);

Is_long

O mesmo que is_int()
Int is_long(mixed var);

Is_object

Retorna true se a variável for um objeto
Int is_object(mixed var);

Is_real

Retorna true se a variável for do tipo real
Int is_real(mixed var);

Is_string

Retorna true se a variável for do tipo string
Int is_string(mixed var);

Isset

Retorna true se uma variável existir
Int isset(mixed var);

Settype

Altera o tipo de uma variável. Retorna true se tiver sucesso; caso contrário, retorna false.
Int settype(string var, string tipo);

Tipos permitidos
Integer
Double
String

Array
Objeto

Strval

Retorna o valor em string de uma variável
String strval(mixed var);

Unset

Exclui uma variável
Int unset(mixed var);

14. Referências na Internet

Site oficial do PHP	http://www.br.php.net http://www.php.net
Site do projeto Zend que originou o PHP4	http://www.zend.org
Site do servidor web Apache, que é amplamente utilizado e compatível com o PHP	http://www.apache.org
Site da lista principal de discussão de PHP no Brasil	http://www.allfinder.com.br/php
Site com diversos exemplos de PHP. Ótimo para iniciantes	http://www.weberdev.com
Site com diversas classes em PHP para utilizar	http://www.thewebmasters.net/php
Site com diversos artigos e tutoriais	http://www.phpbuilder.com
Artigos e informações sobre PHP para WebMasters	http://www.devshed.com/Server_Side/PHP
Tutorial de PHP com MySQL	http://www.hotwired.com/webmonkey/99/21/index2a.html
Como criar um site de busca, como o Yahoo!, em PHP e MySQL.	http://webreference.com/perl/xhoo/php1
Diversos projetos, tutoriais e informações sobre o PHP	http://www.phpwizard.net
PHP Knowledge Base	http://e-gineer.com/phpkb

Universidade Federal de Uberlândia
Faculdade de Computação
PET Sistemas de Informação

Apostila do Curso de PHP Orientado a Objetos

Sumário

O que é o PHP?....4
Como funciona o PHP?....4
O que será necessário para utilizar o PHP neste curso?....4
O arquivo PHP.ini....5
Primeiro programa em PHP Orientado a Objetos: Construindo um Login....7
Classes e Métodos Mágicos....8
Variáveis e métodos....10
Propriedades (atributos ou campos)....11
Variáveis estáticas e métodos estáticos....13
Passagem de parâmetros em métodos....14
Definindo parâmetros default em métodos....15
Herança....17
Interfaces, Classes e métodos abstratos 18
Traits....24
Sobreposição (Overriding)....25
A palavra-chave “final”....26
Clonagem de objetos....27
Comparação de objetos....29
Créditos:....31

Objetivos desta apostila:

Este material foi desenvolvido com o intuito de auxiliar os alunos e professores ao longo do curso de PHP Orientado a Objetos. Presume-se que os alunos deverão possuir conhecimentos de Programação Orientada a Objetos, pois o foco do curso será na aplicação deste estilo de programação na linguagem PHP. Alguns assuntos serão descritos brevemente por se tratar de uma apostila mais de prática de conceitos, porém, existirão várias referências externas para aprofundamento teórico e prático. Esta apostila contém muitas ilustrações para facilitar o aprendizado.

PHP OO

O que é o PHP?

PHP (PHP: Hypertext Preprocessor) é uma linguagem de programação caracterizada por ser bastante utilizada no desenvolvimento de aplicações Web e pela sua facilidade de ser embutida em códigos HTML. PHP permite a criação de páginas dinâmicas, ou seja, que apresentam conteúdo que mudam de acordo com os dados de uma base de dados, por exemplo. É possível coletar dados de formulários e ainda enviar ou receber cookies. PHP é uma linguagem simples para iniciantes, porém, cheia de recursos. Também é possível desenvolver aplicações Desktop utilizando o PHP-GTK.

A versão estável mais atual é a 5.4 (em 03/06/2013). É possível baixar o manual do PHP no link: http://www.php.net/download-docs.php. Para acompanhar lançamentos de novas versões e novidades navegue no site: http://www.php.net/.

Veja a história do PHP em http://www.php.net/manual/pt_BR/history.php.

Como funciona o PHP?

O PHP é interpretado no lado do computador servidor (server-side), ou seja, quando o computador cliente envia uma requisição ao servidor, este último fica responsável por tratá-la de acordo com a necessidade da aplicação: consultar e armazenar dados no banco, processar dados, gerar o HTML que será interpretado e exibido pelo navegador do cliente, dentre outros.

PHP é uma linguagem interpretada, o que quer dizer que não é gerado um código binário executável, portanto, será necessário fornecer o código fonte ao servidor.

O que será necessário para utilizar o PHP neste curso?

Existem várias maneiras de utilizar o PHP, como pode ser visto em http://www.php.net/manual/pt_BR/install.general.php. Para este curso, será utilizado:

- Interpretador PHP;
- Servidor WEB (Apache);
- Browser (Ex.: Google Chrome, Mozilla Firefox, Internet Explorer, Opera, Safari, etc);

- Banco de dados MySQL[1];

É possível hospedar aplicações PHP em hosts gratuitos na WEB, por exemplo, através do Hostinger. Desta maneira, não é necessário a preocupação com instalação e configuração do Apache, PHP e MySQL.

Existem várias combinações de softwares livres que facilitam o processo de instalação e configuração para utilizar o PHP. Exemplos:

- XAMPP (**X** qualquer sistema operacional, **A**pache, **M**ySQL, **P**HP, **P**erl)
- WAMP (**W**indows, **A**pache, **M**ySQL, **P**HP)

Qualquer uma das opções acima contém o pacote de software necessário para desenvolver aplicações em PHP. Portanto, instale um deles para prosseguir as atividades do curso.

Outra ferramenta que facilitará o processo de escrever os scripts PHP é a **IDE** (Integrated Development Environment), um ambiente integrado para desenvolvimento de software. Neste curso será utilizado o software livre NetBeans.

Existem várias fontes de conhecimentos disponíveis em grupos de usuários que respondem dúvidas. O site http://www.phpusergroups.org contém uma lista de grupos de usuários em diversos países, inclusive o Brasil.

O arquivo PHP.ini

O PHP tem um arquivo de configuração chamado *php.ini*. Nele estão definidos várias diretivas que controlam o comportamento do PHP em tempo de execução. Este arquivo é carregado no momento em que o servidor é iniciado, portanto, caso altere alguma das diretivas definidas nele, será necessário que o servidor seja reiniciado.

Vejam na tabela a seguir algumas das diretivas:

1 Não é obrigatório utilizar o MySQL ou qualquer outro banco de dados no desenvolvimento de uma aplicação em PHP, porém, neste curso ele será utilizado em exercícios.

Diretiva	Valor Default	Significado
expose_php	On	Com o valor Off a quantidade de informações disponíveis ao atacante. Ex.: expose_php=Off
date.timezone		Com o valor America/Sao_Paulo está sendo configurado a data e horário corresponde ao de São Paulo para as funções que usam data e horas. Ex.: date.timezone America/Sao_Paulo
disable_functions		É possível desabilitar funções para que o PHP não execute-as. Ex.: disable_functions=phpinfo,exec,proc_open
display_errors	1	Determine se os erros devem ser impresso na tela ou escondido do usuário. 1 imprime, 2 esconde.
file_uploads	1	1 Permite uploads de arquivo via HTTP.
upload_max_filesize	2M	Tamanho máximo de upload arquivo.
max_file_uploads	20	Número máximo de uploads de arquivos simultaneamente
max_execution_time	30	Tempo máximo em segundos que um script PHP pode ser executado.
memory_limit	128M	Define a quantidade de memória que um script pode alocar. Se o valor for -1 não existe limite de memória.
mssql.timeout	60	Tempo máximo da execução de um script no mysql.
post_max_size	8M	Tamanho máximo uma post de dados incluindo upload de arquivos.

Veja uma lista completa de diretivas em http://www.php.net/manual/en/ini.list.php.

Primeiro programa em PHP Orientado a Objetos: Construindo um Login

Abra o NetBeans e crie um novo projeto, acessando **Arquivo → Novo Projeto → PHP → Aplicação PHP**.

Escolha um nome para o projeto, por exemplo: *hellophp*. Certifique que a pasta fonte do projeto esteja no diretório htdocs do XAMPP, por exemplo, *C:\xampp\htdocs\hellophp*.

Clique em finalizar. Após criar o projeto selecione a raiz do projeto e vá em executar ou pressione F6.

Todo código PHP deve estar dentro de um **<?php ?>**. O resultado esperado é que seja aberto uma página em branco no browser padrão do sistema operacional. Caso o mesmo não abra, digite no browser o endereço URI **http://localhost/PASTA RAIZ DO PROJETO/index.php** (Substitua PASTA RAIZ DO PROJETO pela pasta correspondente ao projeto no diretório *htdocs*). Uma página em branco deverá aparecer.

Crie um novo arquivo chamado *usuario.php*.

Classes e Métodos Mágicos

Selecione o arquivo criado e crie uma classe concreta chamada `Usuario`. Nesta classe será criado o método construtor `(__construct)`.

No arquivo index.php instancie a classe que você acabou de criar.

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title></title>
    </head>
    <body>
        <?php
            // put your code here
            include_once 'usuario.php';
            $usuario = new Usuario;//ou new Usuario();
        ?>
    </body>
</html>
```

Desta vez, crie o método destrutor `(__destruct)` da classe `Usuario`.

```
<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
    class Usuario{
        function __construct(){//método construtor
            echo "Sejam Bem Vindos!";
        }
        function __destruct() {//método destrutor
            echo "Bye";
        }
    }
?>
```

Esses dois métodos implementados são chamados pelo PHP de “métodos mágicos” e começam com __ (underscore). Além destes métodos mágicos, existem outros. Veja mais em: http://www.php.net/manual/pt_BR/language.oop5.magic.php

CUIDADO	PHP reserva todas as funções com nomes começando com __ como mágicas. É recomendado que você não use funções com nomes com __ no PHP a não ser que você queira alguma funcionalidade mágica documentada.

NOTA	Não é possível criar dois ou mais métodos construtores no PHP.

Variáveis e métodos

Variáveis em PHP são iniciadas com o símbolo *$*. Além disso, PHP é uma linguagem **não-tipada**. Isto quer dizer que não é necessário declarar o tipo das variáveis, pois o PHP possui a capacidade de inferir sobre tais tipos.

Crie duas variáveis uma chamada `usuario` e a outra `senha`. Em seguida crie os respectivos métodos GETTERS e SETTERS.

DICA	Caso não consiga gerar os métodos GETTERS e SETTERS automaticamente pelo Netbeans, utilize o site http://www.icurtain.co.uk/getset.php, para agilizar na construção dos métodos.

NOTA	Variáveis ou atributos no PHP iniciam com o caractere '$'. `Ex.: $usuario, $senha.`

NOTA	`$this->` é uma pseudo variável utilizada para acessar uma propriedade NÃO estática que está disponível na classe. Ela é uma referência para o objeto que chama inicialmente.

NOTA	É possível definir constantes nas classes PHP. Constantes são valores fixos, ou seja, que não variam ao longo da execução do programa. Veja um exemplo de constante: `const pi= 3.14;` Veja que diferentemente de um atributo da classe, uma constante não possui o símbolo *$*. Para acessar uma constante use: `Classe::pi;` onde `Classe` é o nome da classe.

Agora que você já criou as variáveis `$usuario` e `$senha` com os respectivos métodos GETTERS e SETTERS, teste-os, através do arquivo *index.php* como ilustrado na figura a seguir:

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title></title>
    </head>
    <body>
        <?php
            // put your code here
            include_once 'usuario.php';
            $usuario = new Usuario;//ou new Usuario();
            $usuario->setUsuario("Seu nome");
            echo "<br>".$usuario->getUsuario();
        ?>
    </body>
</html>
```

NOTA	Perceba, que um código HTML ` ` foi posto junto ao PHP. Esta é uma das facilidades que o PHP fornece.

Propriedades (atributos ou campos)

Os métodos e as propriedades (atributos ou campos) possuem visibilidade (contexto) **public**, **protected** e **private** (público, protegido e privado).

Ex.:

Métodos	Atributos	Visibilidade (contexto)

`public function getNome();`
`//ou`
`function getNome();`
	`public $nome;`	Default para os métodos – Em qualquer contexto é possível acessá-lo. *Visibilidade fraca.*

`private function getNome();`	`private $nome;`	Apenas no contexto da classe é possível acessá-lo. *Visibilidade forte.*
`protected function getNome();`	`protected $nome;`	Apenas no contexto da classe e subclasses é possível acessá-lo. *Visibilidade média.*

Caso queira uma explicação mais detalhada e com exemplos, consulte http://www.php.net/manual/pt_BR/language.oop5.visibility.php

Na classe usuário, altere a visibilidade do método `setUsuario()` para `protected` e execute o arquivo *index.php* no Browser.

Página Inicial | index.php | usuario.php

Código-Fonte | Histórico

```
1  <?php
2  class Usuario {
3      private $usuario;   //atributos privados
4      private $senha;
5      function __construct() {//método construtor
6          echo "Sejam Bem Vindos!"; //ou print "Sejam Bem Vindos!";
7      }
8      function __destruct() {//método destrutor
9          echo "Bye";
10     }
11     //GETTERS and SETTERS.
12     //$this é uma pseudo variável que referencia o objeto chamador do método
13     function getUsuario() {
14         return $this->usuario;
15     }
16     protected function setUsuario($_usuario) {
17         $this->usuario = $_usuario;
18     }
19     function getSenha() {
20         return $this->senha;
21     }
22     function setSenha($_senha) {
23         $this->senha = $_senha;
24     }
25 }
26 ?>
```

Ocorrerá o seguinte erro fatal devido não ser permitido chamar um método protegido fora do contexto da classe e das subclasses:

localhost/hellophp/index.php

Sejam Bem Vindos!
Fatal error: Call to protected method Usuario::setUsuario() from context '' in **C:\xampp\htdocs\hellophp\index.php** on line **12**

Altere novamente a visibilidade do método para public.

Variáveis estáticas e métodos estáticos

Crie uma variável estática chamada `$nro_usuarios_online` e atribua o valor 0 para ela. Em seguida, crie um método chamado `alteraNroUsuarios` para que cada vez que o método construtor for chamado, seja chamado o método `alteraNroUsuarios` que por sua vez, incrementará em 1 o número de usuários online.

Página Inicial | index.php | usuario.php

Código-Fonte | Histórico

```
<?php
class Usuario {
    private $usuario;    //atributos privados
    private $senha;
    public static $nro_usuarios_online = 0;

    function __construct() {//método construtor
        echo "Sejam Bem Vindos!"; //ou print "Sejam Bem Vindos!";
        Usuario::alteraNroUsuariosOnline();
    }

    function __destruct() {//método destrutor
        echo "Bye";
    }

    static function alteraNroUsuariosOnline(){
        Usuario::$nro_usuarios_online+= 1;
    }
```

DICA	`self::alteraNroUsuariosOnline()` tem o mesmo efeito de `Usuario::alteraNroUsuariosOnline()`. A palavra reservada `self::` é usada para referenciar elementos estáticos da classe. Enquanto que `$this->` é usado para referenciar elementos não estáticos.

Teste o método criado alterando o arquivo *index.php*.

```
Página Inicial | index.php | usuario.php
Código-Fonte | Histórico

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5          <title></title>
6      </head>
7      <body>
8          <?php
9              // put your code here
10             include_once 'usuario.php';
11             $usuario = new Usuario;//ou new Usuario();
12             $usuario->setUsuario("Seu nome");
13             echo "<br>".$usuario->getUsuario()."<br>";
14
15             echo Usuario::$nro_usuarios_online."<br>";
16             $usuario2 = new Usuario;//ou new Usuario();
17             echo "<br>".Usuario::$nro_usuarios_online."<br>";
18         ?>
19     </body>
20 </html>
```

Para mais exemplos de membros estáticos, consulte o link: http://www.php.net/manual/pt_BR/language.oop5.static.php.

Passagem de parâmetros em métodos

O método `alteraNroUsuariosOnline()` foi feito, porém deseja-se que quando o método `__destruct` seja chamado, seja decrementada em 1 a variável `$nro_usuarios_online`. Uma das soluções, é passar um parâmetro para a função `alteraNroUsuariosOnline()` que permitirá incrementar ou decrementar a variável.

```php
<?php
class Usuario {
    private $usuario;   //atributos privados
    private $senha;
    public static $nro_usuarios_online = 0;

    function __construct() {//método construtor
        echo "Sejam Bem Vindos!"; //ou print "Sejam Bem Vindos!";
        Usuario::alteraNroUsuariosOnline(+1);
    }

    function __destruct() {//método destrutor
        Usuario::alteraNroUsuariosOnline(-1);
        echo "Bye Agora são ".Usuario::$nro_usuarios_online." usuários online!<br />";
    }

    static function alteraNroUsuariosOnline($add_or_remove){
        Usuario::$nro_usuarios_online+= $add_or_remove;
    }
```

NOTA	Embora o método `alteraNroUsuariosOnline($add_or_remove)` tenha recebido um número inteiro como parâmetro, é possível passar qualquer tipo de dados, não somente tipos primitivos (inteiro, float, string, etc.) como também tipos complexos (objetos).

Definindo parâmetros default em métodos

Um recurso interessante do PHP é definir valores default para parâmetros de métodos. No exemplo a seguir, se o programador não passar nenhum parâmetro para o método `alteraNroUsuariosOnline($add_or_remove)`, o valor que esta variável assumirá é 0.

```
Página Inicial | index.php | usuario.php
Código-Fonte | Histórico

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5          <title></title>
6      </head>
7      <body>
8          <?php
9              // put your code here
10             include_once 'usuario.php';
11             $usuario = new Usuario;//ou new Usuario();
12             $usuario->setUsuario("Seu nome");
13             echo "<br>".$usuario->getUsuario()."<br>";
14
15             echo Usuario::$nro_usuarios_online."<br>";
16             $usuario2 = new Usuario;//ou new Usuario();
17             echo "<br>".Usuario::$nro_usuarios_online."<br>";
18                                                 $add_or_remove = 0
19             Usuario::alteraNroUsuariosOnline()
20         ?>
21     </body>
22 </html>
```

Perceba na figura anterior, que é possível utilizar o método `alteraNroUsuariosOnline()` sem passar nenhum parâmetro para a função, pois, o valor default, que no caso é 0, foi definido.

```
Página Inicial | index.php | usuario.php
Código-Fonte | Histórico

1  <?php
2  class Usuario {
3      private $usuario;   //atributos privados
4      private $senha;
5      public static $nro_usuarios_online = 0;
6
7      function __construct() {//método construtor
8          echo "Sejam Bem Vindos!"; //ou print "Sejam Bem Vindos!";
9          Usuario::alteraNroUsuariosOnline(+1);
10     }
11
12     function __destruct() {//método destrutor
13         Usuario::alteraNroUsuariosOnline(-1);
14         echo "Bye Agora são ".Usuario::$nro_usuarios_online." usuários online!<br />";
15     }
16
17     static function alteraNroUsuariosOnline($add_or_remove=0){//parâmetro default 0
18         Usuario::$nro_usuarios_online+= $add_or_remove;
19     }
```

Para mais detalhes sobre como passar argumentos para funções e métodos acesse: http://www.php.net/manual/pt_BR/functions.arguments.php

Herança

No paradigma Orientado a Objetos, vários são os benefícios da herança . Como visto em http://erpbasic.blogspot.com.br/2012/01/inheritance-advantages-and.html, a herança permite reusabilidade de código, extensibilidade, ocultamento de dados, dentre outros. Em PHP, utilizar herança é muito semelhante ao modo de utilizar na linguagem Java. Para isso, é utilizada a palavra reservada `extends`.

Exemplo: no mesmo projeto, adicione uma classe chamada `UsuarioComum` que estenderá a classe `Usuario`.

Na classe `UsuarioComum` existe um método construtor e destrutor que invoca o construtor e destrutor da super classe (`Usuario`). Além disso, um atributo chamado `nro_vezes_acessado` estará na classe para fins estáticos saber quantas vezes um usuário comum acessou o sistema. Certifique de que este atributo será incrementado em um a cada vez que o usuário comum acessar o sistema.

Página Inicial | index.php | usuario.php | usuario_comum.php

Código-Fonte | Histórico

```
1  <?php
2  include_once 'usuario.php';//inclui o arquivo 'usuario.php' --poderia usar include, require ou require_once
3  class UsuarioComum extends Usuario {
4      private $nro_vezes_acessado = 0;   //atributos privados
5
6      function __construct() {//método construtor
7          parent::__construct();//irá chamar o construtor da classe Usuario
8          setNroVezesAcessado($this->nro_vezes_acessado + 1);
9      }
10
11     function __destruct() {//método destrutor
12         parent::__destruct();//irá chamar o destrutor da classe Usuario
13     }
14
15     //GETTERS and SETTERS.
16     //$this é uma pseudo variável que referencia o objeto chamador do método
17     function getNroVezesAcessado() {
18         return $this->nro_vezes_acessado;
19     }
20
21     function setNroVezesAcessado($_nro_vezes_acessado) {
22         $this->nro_vezes_acessado = $_nro_vezes_acessado;
23     }
24 }
25 ?>
```

Questão: É uma boa opção que o atributo `nro_vezes_acessado` seja estático?
Resposta: Não, pois sendo estático, todos os objetos da classe `UsuarioComum` terão acessado o mesmo número de vezes, gerando assim um erro semântico.

Para testar a classe criada, insira no `index.php` o seguinte código:

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title></title>
    </head>
    <body>
        <?php
            // put your code here
//            include_once 'usuario.php';
//            $usuario = new Usuario;//ou new Usuario();
//            $usuario->setUsuario("Seu nome");
//            echo "<br>".$usuario->getUsuario()."<br>";
//
//            echo Usuario::$nro_usuarios_online."<br>";
//            $usuario2 = new Usuario;//ou new Usuario();
//            echo "<br>".Usuario::$nro_usuarios_online."<br>";
//
//            Usuario::alteraNroUsuariosOnline();
//-----------------------------HERANÇA-----------------------------------
            include_once 'usuario_comum.php';
            $usuario = new UsuarioComum;
            $usuario->setUsuario("PET-SI");
            echo "<br>".$usuario->getUsuario()."<br>";
            echo "Usuarios online ".Usuario::$nro_usuarios_online."<br>";
        ?>
    </body>
</html>
```

Houve algum erro? Pense em como corrigir isso. Uma dica: `this` is the problem.

Exercício: Implemente uma classe `Endereco` que estenda a classe `Empregado`. O `Empregado` deve ter um ID, nome, sexo e um salário. A classe `Endereco` deve conter, além dos atributos da classe mãe, os atributos país, cidade, rua e número. As classes devem conter implementados em si os conceitos de encapsulamento (atributos privados e métodos de manipulação públicos).

Interfaces, Classes e métodos abstratos

Classes abstratas são classes que não podem ser instanciadas, isto é, não é possível criar objeto da classe abstrata.

Caso um método de uma classe for abstrato, obrigatoriamente a classe será abstrata.

Um método abstrato possui apenas a sua assinatura (especificação) na classe abstrata, isto "obriga" que as subclasses (classe filha) o implemente. Ele ainda deve possuir visibilidade pública ou protegida.

Uma classe abstrata pode conter métodos abstratos (sem implementação) e métodos não abstratos.

Uma interface contém apenas métodos abstratos.

A classe filha define o método com mesma ou mais fraca visibilidade.

Para praticar, a classe `Usuario` será uma *abstract class*. Além disso, será adicionado o método abstrato `logar()`.

```
<?php
abstract class Usuario {
    private $usuario;   //atributos privados
    private $senha;
    public static $nro_usuarios_online = 0;

    abstract protected function logar();

    function __construct() {//método construtor
        echo "Sejam Bem Vindos!"; //ou print "Sejam Bem Vindos!";
        Usuario::alteraNroUsuariosOnline(+1);
    }

    function __destruct() {//método destrutor
        Usuario::alteraNroUsuariosOnline(-1);
        echo "Bye Agora são ".Usuario::$nro_usuarios_online." usuários online!<br />";
    }

    static function alteraNroUsuariosOnline($add_or_remove=0){//parâmetro default 0
        Usuario::$nro_usuarios_online+= $add_or_remove;
    }...
```

Devido à classe `UsuarioComum` ser uma subclasse da classe abstrata `Usuario`, será necessário que a classe `UsuarioComum` implemente o método `abstract logar()`.

```
1  <?php
2  include_once 'usuario.php';//inclui o arquivo 'usuario.php' --poderia usa
3  class UsuarioComum extends Usuario {
4      private $nr
5      
       function __
7          parent::__construct();//irá chamar o construtor da classe Usuario
8          $this->setNroVezesAcessado($this->nro_vezes_acessado + 1);
9      }
```

\UsuarioComum não é abstrata e não substitui o método abstrato logar() em \Usuario
---
(Alt-Enter mostra dicas)

Portanto, para implementar automaticamente o método logar no NetBeans, coloque o cursor do mouse na linha que está ocorrendo o erro (no caso linha 3) e tecle ALT+ENTER:

```
<?php
include_once 'usuario.php';//inclui o arquivo 'us
class UsuarioComum extends Usuario {
    //atributo
```

- Implementar Todos os Métodos Abstratos
- Declarar Classe Abstrata
- Personalizar Dicas na Caixa de Diálogo Opções

Selecione a alternativa Implementar Todos os Métodos Abstratos.

Teste a aplicação rodando o arquivo index.php. Em seguida, experimente descomentar as linhas 10 e 11 correspondentes na figura a seguir:

NOTA	Uma classe pode *estender* apenas uma classe.

```
Página Inicial | index.php | usuario.php | usuario_comum.php
Código-Fonte | Histórico

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5          <title></title>
6      </head>
7      <body>
8          <?php
9              // put your code here
10             include_once 'usuario
11             $usuario = new Usuario;//ou new Usuario();
12 //             $usuario->setUsuario("Seu nome");
13 //             echo "<br>".$usuario->getUsuario()."<br>";
14 //
15 //             echo Usuario::$nro_usuarios_online."<br>";
16 //             $usuario2 = new Usuario;//ou new Usuario();
17 //             echo "<br>".Usuario::$nro_usuarios_online."<br>";
18 //
19 //             Usuario::alteraNroUsuariosOnline();
20 //-----------------------------HERANÇA-----------------------------------
21             include_once 'usuario_comum.php';
22             $usuario = new UsuarioComum;
23             $usuario->setUsuario("PET-SI");//usando métodos da Super classe
24             echo "<br>".$usuario->getUsuario()."<br>";
25             echo "Usuarios online ".Usuario::$nro_usuarios_online."<br>";
26         ?>
27     </body>
28 </html>

A classe abstrata \Usuario não pode ser instanciada
---
(Alt-Enter mostra dicas)
```

Note a importância do conhecimento dos conceitos teóricos na compreensão dos erros que acontecem na prática.

Exercício: Crie uma classe chamada `UsuarioAdmin`, que herda da classe `Usuario`.

Para exemplificar o uso de interfaces, crie uma interface chamada `IUsuario`, que contém o método `logar (Usuario u)`. `u` é um objeto da classe `Usuario`.

```
Página Inicial | index.php | usuario.php | usuario_comum.php | usuario_admin.php | iusuario.php
Código-Fonte | Histórico

1  <?php
2
3      interface IUsuario{
4          //Uma interface obriga a implementação dos métodos dela nas classes que implementam a interface IUsuario.
5          public function imprimir_dados(Usuario $u);//ou simplesmente $u.
6      }
7  ?>
```

Quando é definido que o método `imprimir_dados (Usuario u)` recebe um objeto `u` da classe `Usuario`, significa que o método está sendo forçado a receber um objeto deste tipo. Isto é denominado *Indução de tipo.*

Feito isso, a imagem a seguir ilustra a classe abstrata `Usuario` implementando a interface `IUsuario`.

```
<?php
include_once 'iusuario.php';
abstract class Usuario implements IUsuario{
    private $usuario;   //atributos privados
    private $senha;
    public static $nro_usuarios_online = 0;

    abstract protected function logar();

    //Usuario tem a opção de implementar o método imprimir_dados ou deixar que as subclasses dela
                                                                                        o implemente.
    //public abstract function imprimir_dados(Usuario $u);  // deixa que as subclasses dela
                                                            //o implemente(deixando abstract).  1

    //Também é um exemplo de polimorfismo!
    public function imprimir_dados(Usuario $u){ //implementa  2
       echo "Classe usuário: ".get_class($u)."<br>";
    }...
```

Veja na imagem, que existem duas maneiras de tratar o método `imprimir_dados(Usuario $u)` que foi definido em `IUsuario`:

1) O fato de a classe `Usuario` ser abstrata possibilita que ela deixe que esse método seja implementado pelas subclasses dela.

2) Implementa o método eliminando a necessidade de que as subclasses implemente-os.

No arquivo *index.php* faça com que um objeto do tipo `Usuario` chame o método `imprimir_dados(Usuario $u)`.

Página Inicial | index.php | usuario.php | usuario_comum.php | usuario_admin.php

Código-Fonte | Histórico

```
5            <title></title>
6        </head>
7        <body>
8            <?php
9   //           include_once 'usuario.php';
10  //           $usuario = new Usuario;//ou new Usuario();
11  //           $usuario->setUsuario("Seu nome");
12  //           echo "<br>".$usuario->getUsuario()."<br>";
13  //
14  //           echo Usuario::$nro_usuarios_online."<br>";
15  //           $usuario2 = new Usuario;//ou new Usuario();
16  //           echo "<br>".Usuario::$nro_usuarios_online."<br>";
17  //
18  //           Usuario::alteraNroUsuariosOnline();
19  //---------------------HERANÇA-INTERFACE----------------------------------------
20               include_once 'usuario_comum.php';
21               $usuario = new UsuarioComum;
22               $usuario->setUsuario("PET-SI");//usando métodos da Super classe
23               echo "<br>".$usuario->getUsuario()."<br>";
24               echo "Usuarios online ".Usuario::$nro_usuarios_online."<br>";
25               $usuario->imprimir_dados($usuario);//chamando método definido na interface
26           ?>
27       </body>
28   </html>
```

NOTA	Classes abstratas e concretas podem implementar várias interfaces.

NOTA	Interfaces podem definir constantes. Constantes são uma espécie de variável e cujo o seu valor não é alterado durante a execução do programa. Ex.:

```
<?php
    interface IUsuario{
        //Uma interface obriga a implementação dos
        public function imprimir_dados(Usuario $u)
        const curso = "CURSO PHP - PET-SI";
    }
?>
```

Interfaces podem “extender” outras classes:

```php
<?php interface IFoo
        {
            public function Foo();
        }

        interface IBar extends IFoo
        {
            public function Bar($bar);
        }

        class FooBar implements IBar
        {
            public function Foo()
            {
                echo "Foo";
            }

            public function Bar($bar)
            {
                echo $bar;
            }
        }
        $IFoo = new FooBar;
        $IFoo->Foo();
        $IFoo->Bar("teste");
?>
```

Exercício: Crie uma interface no projeto que declara pelo menos um método. Em seguida, mostre um exemplo de como utilizá-la.

Traits

Trait é mais um mecanismo de reúso de código. O PHP não implementa herança múltipla, por isso, uma maneira de reduzir esta limitação é por meio do uso de traits. Desta maneira, o desenvolvedor é capaz de reusar um conjunto de métodos situados em classes independentes que estão em outra hierarquia.

Um trait é semelhante a uma classe, porém, com um grupo de funcionalidades. Não é possível instanciá-lo como uma classe concreta.

```php
<?php

trait Hello {
    public function sayHello() {
        echo "Olá";
    }
}

trait World {
    public function sayWorld() {
        echo "Mundo";
    }
}

class View {
    use Hello, World;
}

$view = new View();
$view->sayHello();
$view->sayWorld();
// exibe: Olá Mundo

?>
```

Neste exemplo, utilizando traits é possível que a classe `View` use métodos dos traits `Hello` e `World`, o que seria impossível utilizando herança simples.

É possível realizar outras coisas com traits, como por exemplo, mudar a visibilidade de um método Para saber mais sobre traits consulte: http://www.php.net/manual/pt_BR/language.oop5.traits.php

DICA	Utilize vírgulas para separar os traits a serem utilizados, como mostrado no exemplo da imagem (`use Hello, World;`).

Sobreposição (Overriding)

É possível sobrepor métodos no PHP de maneira que o método em uma subclasse seja invocado ao invés do método de sua superclasse. Veja na imagem a seguir um exemplo de sobreposição do método `imprimir_dados(Usuario $u)` da classe `usuarioComum`. Este método também é um considerado polimórfico, já que ele é herdado e sobreposto da superclasse e ainda, cada subclasse que implemente este método poderá dar um comportamento diferente para o mesmo método.

```
<?php
include_once 'usuario.php';//inclui o arquivo 'usuario.php' --poderia usar include, require ou requi
class UsuarioComum extends Usuario {
    •
    •
    •
    //sobrepondo o método imprimir_dados(Usuario $u) da classe usuario
    public function imprimir_dados(Usuario $u){
        //echo "Classe usuário: ".get_class($u)."<br>";
        echo "Método chamado na Classe Filho <br />";
    }
```

Execute o *index.php*.

A palavra-chave "final"

A palavra-chave `final` previne que classes filhas sobreponham um método mais acima na hierarquia. Deste modo, torna-se impossível criar métodos de mesmo nome derivados da classe que utiliza a palavra-chave `final`.

```
<?php
class ClasseBase {
   public function teste() {
       echo "ClasseBase::teste() chamado\n";
   }

   final public function maisTeste() {
       echo "ClasseBase::maisTeste() chamado\n";
   }
}

class ClasseFilha extends ClasseBase {
   public function maisTeste() {
       echo "ClasseFilha::maisTeste() chamado\n";
   }
}
?>
```

O código anterior resulta em um erro fatal pois `ClasseFilha` tenta realizar a sobreposição do método `maisTeste` de `ClasseBase`, que por sua vez está definido utilizando a palavra-chave `final`, que não permite que métodos de sobreposição sejam criados para o método inicial.

Assim como métodos, as classes que utilizam a palavra-chave `final` também não permitem que outras classes herdem seus atributos e métodos, o que resultaria em erro, como o código do exemplo abaixo:

```
<?php
final class ClasseBase {
   public function teste() {
       echo "ClasseBase::teste() chamado\n";
   }

   final public function maisTeste() {
       echo "ClasseBase::maisTeste() chamado\n";
   }
}

class ClasseFilha extends ClasseBase {

}
?>
```

Pratique:

1. Na classe `UsuarioComum` faça a sobreposição do método imprimir_dados(`Usuario $u`).
2. Na classe `Usuario` coloque a palavra reservada final no método `imprimir_dados(Usuario $u)`. Certifique de que entendeu o conceito e em seguida deixe o método `imprimir_dados(Usuario $u)` sem a palavra final.

Clonagem de objetos

Clonagem de objetos é utilizada quando se deseja criar uma cópia de um objeto, ao invés, de ter a referência dele propriamente dita. A clonagem de um objeto é feita através da palavra-chave `clone`. Ex.: `$copia_do_objeto = clone $objeto;`.

Veja o exemplo a seguir que ilustra o arquivo *index.php* mostra diferença entre usar clonagem de objetos e usar a cópia da referência do objeto:

```php
include_once 'usuario_comum.php';
$usuario = new UsuarioComum;
$usuario->setUsuario("PET-SI");//usando métodos da Super classe
//echo "<br>".$usuario->getUsuario()."<br>";
//echo "Usuarios online ".Usuario::$nro_usuarios_online."<br>";
//$usuario->imprimir_dados($usuario);//chamando método definido na interface

//clonagem de objetos
$copia_referencia_usuario = $usuario;
$clone_usuario = clone $usuario;

$usuario->setUsuario("UFU");

echo "<br>".$usuario->getUsuario()."<br>";                //Imprimirá UFU
echo $copia_referencia_usuario->getUsuario()."<br>";     //Também Imprimirá UFU!!!
echo $clone_usuario->getUsuario()."<br>";                //Imprimirá PET-SI
```

localhost/hellophp/index.php

Sejam Bem Vindos!
UFU
UFU
PET-SI

```php
echo "<br>".$usuario->getUsuario()."<br>";                //Imprimirá UFU
echo $copia_referencia_usuario->getUsuario()."<br>";     //Também Imprimirá
echo $clone_usuario->getUsuario()."<br>";                //Imprimirá PET-SI
```

Bye Agora são 0 usuários online!
Bye Agora são -1 usuários online!

-1 usuários online! ???

CUIDADO	Veja que na figura anterior imprimiu "-1 usuários online!". Isto ocorre porque o clone não invoca o método `__construct` do objeto `UsuarioComum`, que adicionaria em 1 o número de usuários, porém invoca o método `__destruct` que decrementa em 1 o número de usuários. Para contornar este problema, uma solução será mostrada a seguir.

Na classe UsuarioComum foi implementado o método mágico __clone de maneira que ele invoca o método construtor.

```
...nicia  index.php  usuario_comum.php  usuario_admin.php  usuario.php  iusuario.php
Código-Fonte   Histórico

1  <?php
2  include_once 'usuario.php';//inclui o arquivo 'usuario.php' --poderia usar
3  class UsuarioComum extends Usuario {
4      private $nro_vezes_acessado = 0;    //atributos privados
5
6      function __construct() {//método construtor
7          parent::__construct();//irá chamar o construtor da classe Usuario
8          $this->setNroVezesAcessado($this->nro_vezes_acessado + 1);
9      }
10
11     //método mágico chamado quando se faz um clone de um objeto
12     function __clone() {
13         self::__construct();
14     }
15
16     function __destruct() {//método destrutor
17         parent::__destruct();//irá chamar o destrutor da classe Usuario
18     }
```

Comparação de objetos

Existem duas maneiras de comparar objetos:

- `($objeto1 == $objeto2)` e `(!=)`: verifica se dois objetos são da mesma classe e possuem os mesmos atributos e valores.
- `($objeto1 === $objeto2)` e `(!==)`: verifica se dois objetos são da mesma classe, possuem os mesmos atributos e valores e representam a **mesma instância da classe**.

Alterando o arquivo *index.php* para realizar a comparação entre o objeto `$usuario`, `$clone_usuario` e `$copia_referencia_usuario`.

```
22                $usuario->setUsuario("PET-SI");//usando métodos da Super classe
23                //echo "<br>".$usuario->getUsuario()."<br>";
24                //echo "Usuarios online ".Usuario::$nro_usuarios_online."<br>";
25                //$usuario->imprimir_dados($usuario);//chamando método definido na interface
26
27                //clonagem de objetos
28                $copia_referencia_usuario = $usuario;
29                $clone_usuario = clone $usuario;
30
31    //            $usuario->setUsuario("UFU");
32    //            echo "<br>".$usuario->getUsuario()."<br>";              //Imprimirá UFU
33    //            echo $copia_referencia_usuario->getUsuario()."<br>";    //Também Imprimirá UFU!!!
34    //            echo $clone_usuario->getUsuario()."<br>";               //Imprimirá PET-SI
35
36                //comparação de objetos
37                if ($usuario == $copia_referencia_usuario)
38 1                echo "<br>Usuário tem as mesmas propriedades da copia de sua referência"."<br>";
39                else
40                    echo "Usuário não tem as mesmas propriedades da copia de sua referência"."<br>";
41                if ($usuario == $clone_usuario)
42 2                echo "Usuário tem as mesmas propriedades de seu clone"."<br>";
43                else
44                    echo "Usuário não tem as mesmas propriedades da copia de seu clone"."<br>";
45
46                if ($usuario === $copia_referencia_usuario)
47 3                echo "Usuário é a mesma instância da copia de sua referência"."<br><br>";
48                else
49                    echo "Usuário não é a mesma instância da copia de sua referência"."<br><br>";
50                if ($usuario === $clone_usuario)
51 4                echo "Usuário é a mesma instância do seu clone"."<br>";
52                else
53                    echo "Usuário não é a mesma instância do seu clone"."<br>";
```

O resultado será para cada sequência de `if` e `else`:

1- Usuário tem as mesmas propriedades da cópia de sua referência
2- Usuário não tem as mesmas propriedades da cópia de seu clone
3- Usuário é a mesma instância da cópia de sua referência
4- Usuário não é a mesma instância do seu clone

Isso mostra mais uma vez a diferença entre clonar um objeto e copiar a referência de um objeto.

Créditos:

Higor Ernandes Ramos Silva
Leonardo Pimentel Ferreira

Curso Completo de PHP – PDO

Sejam todos bem vindos (as) ao curso de PDO – PHP Data Objects

Requisito: Para começar o curso de PDO, aconselho que veja antes, todas as aulas do curso de **HTML**, php, Banco de dados MYSQL.

Título das Aulas + links abaixo:

Curso PDO – Aula 01 – Introdução
Curso PDO – Aula 02 – Conexão com PDO
Curso PDO – Aula 03 – Selecionando Dados com PDO
Curso PDO – Aula 04 – Cadastrando Dados com PDO
Curso PDO – Aula 05 – Deletando Dados com PDO
Curso PDO – Aula 06 – Atualizando Dados com PDO

Aula 01 – Introdução

Hoje iremos começar nosso curso exclusivo de PDO (PHP Data Objects).

O PDO vem dominando o mercado no mundo dos programadores PHP, e por este motivo não pode ficar para trás correto?

Neste curso iremos aprender a fazer um CRUD (create | read | update | delete – criar | ler | atualizar | deletar) completo com PDO, e mais. Iremos também fazer comparações com o modo primitivo do PHP, assim todos nós iremos atender as diferenças entre ambos.

Qual a vantagem do uso do PDO?

Imagine que você criou um sistema utilizando o modo primitivo do php, e o tipo de banco de dados utilizado foi o MYSQL.

Depois de muito tempo, seu cliente, fala pra você que o banco de dados agora será do tipo ORACLE.

Particularmente, você teria de mudar diversas páginas correto?

Com o PDO já é diferente, pois o próprio trabalha com driver's de banco de dados, ou seja, para solucionar o problema de seu cliente utilizando PDO, bastaria modificar apenas o arquivo de conexão com o banco de dados, informando o driver desejado.

Com PDO, podemos ficar mais tranquilos co-relação a vulnerabilidades , pois com a utilização do próprio , poderemos evitar situações de SQL Injection.

Está gostando? Continue acompanhando o curso, pois este é apenas o inicio.

Gostaria de comprar um curso online conosco, e não conhece nosso campus virtual? Visite: http://www.escolacompleta.com/campus

Continuação… vamos a uma rápida introdução do PDO.

Introdução

O PHP Data Objects (PDO) define uma interface leve em PHP. Cada driver de banco de dados que implementa a interface PDO pode expor dados de recursos específicos. Note que você não pode executar quaisquer funções de banco de dados usando a extensão PDO, por si só, você deve usar um banco de dados específico do driver PDO para acessar um servidor de banco de dados.

PDO fornece um acesso de dados abstrato, o que significa que, independentemente de qual banco de dados você está usando, você usa as mesmas funções para emitir consultas e buscar dados (CRUD).

Aula 02 – Conexão com PDO

Nesta aula iremos aprender a criar uma conexão com o banco de dados utilizando o PDO, lembrando que faremos comparações do modo primitivo e o modo PDO.

De fato uma conexão primitiva sem a utilização do PDO, ficaria como o exemplo abaixo:

```
<?php
    $host = "localhost";

    $user = "root";
    $pass = "";

    $bd = "banco_teste";


    $conexao = mysql_connect($host,$user,$pass) OR DIE ("Erro ao conectar com o
banco de dados: ".mysql_error());


    if($conexao){
        mysql_select_db($bd) OR DIE ("Erro ao selecionar banco de dados:
".mysql_error());

    }else{}
?>
```

De certa forma o modo primitivo é menos complexo, e mais rápido de ser desenvolvido, contudo existem desvantagens.
Uma delas é a vulnerabilidade;
Outra questão é o famoso código macarrão;
E mais uma é a questão de tempo longo para mudanças de tipos de banco de dados.
Veja agora como criar sua conexão com banco de dados utilizando o PDO, baseando-se no exemplo acima.

```
<?php
    $host = "localhost";

    $user = "root";
    $pass = "";

    $bd = "banco_teste";
    /*O que o PDO pede na conexao?

        ex: PDO($dsn,$user,$pass);
        mais o que é dsn ?

        dsn = tipo_do_banco:host='seu_host';dbname='nome_do_seu_banco';
    */

    $dsn = "mysql:host=".$host.";dbname=".$bd;
    try{

    $conexao = new PDO($dsn,$user,$pass);
    $conexao->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    return $conexao;
    }catch(PDOException $error){

         echo "Erro ao conectar com o banco de dados: ".$error->getMessage();
```

```
    }

?>
```

O que o PDO pede de diferente na conexão?

ex: PDO($dsn,$user,$pass);

Como funciona a estrutura do dsn ?

A estrutura do dsn funciona da seguinte forma:
ex: $dsn = "tipo_do_banco:host=seu_host;dbname=nome_do_seu_banco";
resultado: $dsn = "mysql:host=localhost;dbname=banco_teste";
Até ai tudo bem… mas o que quer dizer está linha:

```
$conexao->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

setAttribute – Define um atributo na declaração. Para ver a lista de atributos clique aqui.
PDO::ATTR_ERRMODE – relatório de erros.
PDO::ERRMODE_EXCEPTION - Atributo filho do atributo ATTR_ERRMODE , este representa um erro gerado pelo PDO.

Endentendo os atributos

Quando você define um atributo no PDO, os próprios possuem valores padrões, e filhos únicos.
A sintaxe para a utilização funciona da seguinte forma:
setAttribute(atributo, filho_do_atributo);
Para isso, veja a lista abaixo , que seta os atributos , e seus filhos alheios.

1. PDO::ATTR_CASE – nomes de colunas força para um caso específico.
    - Filhos do atributo PDO::ATTR_CASE
    - PDO::CASE_LOWER – Força nomes de coluna para minúsculas.
    - PDO::CASE_NATURAL – nomes de colunas deixar como retornados pelo driver de banco de dados.
    - PDO::CASE_UPPER – nomes de colunas força para maiúsculas.
2. PDO:: ATTR_ERRMODE – relatório de erros.
    - Filhos do atributo PDO :: ATTR_ERRMODE
    - PDO::ERRMODE_SILENT – Basta definir códigos de erro.
    - PDO::ERRMODE_WARNING – Levante E_WARNING.
    - PDO::ERRMODE_EXCEPTION – Throw exceções.
3. PDO::ATTR_ORACLE_NULLS – (disponível com todos os drivers, e não apenas da Oracle): Conversão de strings nulas e vazias.
    - Filhos do atributo PDO :: ATTR_ORACLE_NULLS
    - PDO::NULL_NATURAL – Não conversão.
    - PDO::NULL_EMPTY_STRING – string vazio é convertido para NULL .
    - PDO::NULL_TO_STRING – NULL é convertido para uma cadeia vazia.

Existem muitos atributos e seus alheios , para isso saiba mais no site do oficial do php clicando aqui.
Agora que sabemos mais sobre os atributos devemos saber sobre está linha:

```
}catch(PDOException $error){
```

O que é o PDOException?
Ele representa um erro gerado pelo PDO.
Desta forma estamos atribuindo o erro gerado pelo PDO à variável $error. Desta forma devemos apenas chamar a mensagem de error utilizando a variável $error

Observem que para mostrar a mensagem de error gerada, eu utilizei:

```
$error->getMessage();
```

Mas o que é o getMessage()?
getMessage() — Obtém a mensagem da exceção.

Aula 03 – Selecionando Dados com PDO

Nesta aula iremos aprender a fazer a seleção, ou listagem de dados com PDO.
Lembrando que iremos fazer as comparações e observações devidas com o modo primitivo do PHP.
Primeiramente temos o primeiro exemplo, onde iremos fazer a seleção dos dados utilizando o modo primitivo do php.
Veja o exemplo abaixo:

```
< ?php
        include_once("conexao.php"); /*incluindo arquivo de conexao da aula
passada*/

        $sql = "SELECT * FROM alunos";
        $query = mysql_query($sql);


        if(mysql_num_rows($query) > 0){

            while($linha_alunos = mysql_fetch_array($query)){
                    echo $linha_alunos['nome_aluno']."<br />";

            }
        }else{}

?>
```

Simples não?
Lembrando que fizemos a inclusão do arquivo de conexão da aula passada.
Se você não visitou a aula passada clique aqui.
Agora que você já sabe como funciona a seleção de dados utilizando o modo primitivo, iremos agora fazer o mesmo utilizando o PDO.
Primeiramente iremos fazer uma seleção comum sem a utilização de condições (WHERE).

SELEÇÃO COMUM COM PDO

```
< ?php
    include_once("conexao.php");

    function listar_alunos(){
    try{

        $listar = $conexao->query("SELECT * FROM alunos");
        $listar->execute();


            if($listar->rowCount() > 0){

                return $listar->fetchAll(PDO::FETCH_ASSOC);
```

```
            }else{

             echo "Não existem alunos cadastrados na base.";
            }

    }catch(PDOException $error){
        echo "Error ao listar alunos: ".$error->getMessage();

    }
    }

    $alunos = listar_alunos();


    foreach($alunos as $aluno):
            echo $aluno['nome_aluno']."<br />";

    endforeach;
?>
```

Você poderia também, utilizar o FETCH_OBJ ao invés de FETCH_ASSOC , contudo nosso foreach ficaria desta forma:

```
foreach($alunos as $aluno):
        echo $aluno->nome_aluno."<br />";

endforeach;
```

O resultado de todo o código utilizando FETCH_OBJ seria:

```
                                        < ?php
    include_once("conexao.php");//inclusao do arquivo de conexao da aula
passada

    function listar_alunos(){
    try{

        $listar = $conexao->query("SELECT * FROM alunos");
        $listar->execute();


            if($listar->rowCount() > 0){

                return $listar->fetchAll(PDO::FETCH_OBJ);
            }else{

             echo "Não existem alunos cadastrados na base.";
            }

    }catch(PDOException $error){
        echo "Error ao listar alunos: ".$error->getMessage();

    }
    }
```

```
    $alunos = listar_alunos();


    foreach($alunos as $aluno):
            echo $aluno->nome_aluno."<br />";

    endforeach;
?>
```

<h3>SELEÇÃO COMPOSTA COM PDO</h3>

A seleção composta já é diferente, pois iremos utilizar condições , e devemos utilizar o prepare do PDO.
O método query é utilizado apenas para seleções comuns que não envolvem condições (WHERE).

Exemplo de seleção composta:

```
<?php
    include_once("conexao.php");//inclusao do arquivo de conexao da aula
passada

    function listar_alunos_com_condicao(){
    try{

        $idade = 15;/*aqui defini a idade desejada para condicao*/
        $listar = $conexao->prepare("SELECT * FROM alunos WHERE idade = :idade");

        $listar->bindValue(":idade",$idade);/*O bindValue protege os dados que
você está passando , seta o valor para o parametro.*/
        $listar->execute();


            if($listar->rowCount() > 0){

                return $listar->fetch(PDO::FETCH_ASSOC);
            }else{

                  echo "Não existem alunos cadastrados na base.";
            }

    }catch(PDOException $error){
        echo "Error ao listar alunos: ".$error->getMessage();

    }
    }

    $alunos = listar_alunos_com_condicao();


    foreach($alunos as $aluno):
            echo $aluno['nome_aluno']."<br />";
```

```
    endforeach;
?>
```

Observem que no exemplo acima, eu não utilizei o fetchAll, mas sim apenas o fetch, pois estou querendo retornar apenas um valor específico informado em minha "condição".
Lembrando também que ao invés de utilizar bindValue, podemos utilizar também bindParam, que irá retornar o mesmo valor , contudo você poderá também especificar o tipo do parâmetro (type) passado e o tamanho (lenght) .
Exemplo:

```
$listar->bindParam(":idade",$idade,PDO::PARAM_INT,3);
```

Desta forma estou especificando que o valor passado é do tipo inteiro (PARAM_INT), e o tamanho é 3. Ou seja, não existe pessoa com idade maior que 3 digitos correto? (Por favor, não inclua as múmias neste rolo).

E para tipos strings? Como seria?

```
$listar->bindParam(":aluno_nome",$aluno_nome,PDO::PARAM_STR, 255);
```

Já no exemplo acima, estou informando que o parâmetro aluno_nome é do tipo string(PARAM_STR) e o tamanho máximo é de 255 caracteres.
Caso queiram saber mais sobre bindParam's clique aqui
Agora, irei mostrar uma situação que muitos alunos têm dúvida. O famoso ponto de interrogação.
Quando passamos parâmetros, podemos também utilizar o ponto de interrogação, e logo após a posição do índice que pretendemos atribuir o valor corrente.
Exemplo:

```
<?php
    include_once("conexao.php");

    function listar_alunos_com_condicao(){
    try{

        $idade = 15;
        $listar = $conexao->prepare("SELECT * FROM alunos WHERE idade = ?");

        $listar->bindValue(1,$idade);
        $listar->execute();


            if($listar->rowCount() > 0){

                return $listar->fetch(PDO::FETCH_ASSOC);
            }else{

             echo "Não existem alunos cadastrados na base.";
            }

    }catch(PDOException $error){
        echo "Error ao listar alunos: ".$error->getMessage();

    }
```

```
    }

    $alunos = listar_alunos_com_condicao();


    foreach($alunos as $aluno):
            echo $aluno['nome_aluno']."<br />";

    endforeach;
?>
```

Vejam que as linhas modificadas foram as listadas abaixo:

```
$listar = $conexao->prepare("SELECT * FROM alunos WHERE idade = ?");

    $listar->bindValue(1,$idade);/*O bindvalue protege os
```

Contudo, fica a pergunta. E se eu quisesse passar dois parâmetros na condição? Como ficaria?

Veja o exemplo abaixo:

```
<?php
    include_once("conexao.php");

    function listar_alunos_com_condicao(){
    try{

        $idade = 15;
        $sobrenome = "Alberto";

        $listar = $conexao->prepare("SELECT * FROM alunos WHERE idade = ? AND
sobrenome = ?");
        $listar->bindValue(1,$idade);

        $listar->bindValue(2,$sobrenome);
        $listar->execute();


            if($listar->rowCount() > 0){

                return $listar->fetch(PDO::FETCH_ASSOC);
            }else{

             echo "Não existem alunos cadastrados na base.";
            }

    }catch(PDOException $error){
        echo "Error ao listar alunos: ".$error->getMessage();

    }
    }

    $alunos = listar_alunos_com_condicao();
```

```
    foreach($alunos as $aluno):
            echo $aluno['nome_aluno']."<br />";

    endforeach;
?>
```

Concluindo nosso exemplo, vejam as linhas alteradas.

```
$sobrenome = "Alberto";
$listar = $conexao->prepare("SELECT * FROM alunos WHERE idade = ? AND sobrenome =
?");

$listar->bindValue(1,$idade);/*O bindvalue protege os dados que você está
passando.*/
$listar->bindValue(2,$sobrenome);
```

Viram como é simples fazer seleção com o PDO?

Aula 04 – Cadastrando Dados com PDO

Nesta aula iremos aprender a fazer cadastros com o PDO.
Primeiramente veremos um exemplo utilizando o modo primitivo do php.

CADASTRANDO SEM UTILIZAR PDO

```
<?php
    include_once("conexao.php");/*Inclui arquivo de conexao com o bd*/

$nome = "Carlos";
$sobrenome = "Alberto";


$sql = "INSERT INTO alunos (nome,sobrenome) VALUES ('$nome','$sobrenome')";

$query = mysql_query($sql);


    if($query){
        echo "Aluno cadastrado com sucesso!";

    }else{
        echo "Erro ao cadastrar aluno.";

    }
?>
```

Vejam que o cadastro do primitivo fica totalmente vulnerável.
Veja agora, o exemplo acima utilizando PDO.

CADASTRANDO UTILIZANDO PDO

```
<?php
    include_once("conexao.php");
```

```
    $nome = "Carlos";
    $sobrenome = "Alberto";


    try{

        $cadastrar = $conexao->prepare("INSERT INTO alunos (nome,sobrenome)
                        VALUES (:nome,:sobrenome)";

        $cadastrar->bindValue(":nome",$nome);
        $cadastrar->bindValue(":sobrenome",$sobrenome);

        $cadastrar->execute;
        if($cadastrar->rowCount() > 0){

            echo "Aluno cadastrado com sucesso!";
        }else{

            echo "Erro ao cadastrar aluno.";
        }

    }catch(PDOException $error){
        echo "Erro ao cadastrar aluno: ". $error->getMessage();

    }
?>
```

Lembrando, que com a utilização do PDO, eu não precisei deixar o parâmetro dentro de aspas simples, como foi utilizado no modo primitivo.
Veja:

```
$cadastrar = $conexao->prepare("INSERT INTO alunos (nome,sobrenome)
                VALUES (:nome,:sobrenome)";
```

Vimos também que utilizei uma condicional diferente, veja:

```
if($cadastrar->rowCount() > 0){
        echo "Aluno cadastrado com sucesso!";

    }else{
        echo "Erro ao cadastrar aluno.";

    }
```

Fica a pergunta: Mas o rowCount não funciona apenas para selecionar dados?

Resposta: Não! O rowCount , neste caso está com a função do antigo mysql_affected_rows, que retorna o número de linhas afetadas.
Em nosso caso, 1 cadastrado foi executado com sucesso , resultando então na mensagem “Aluno cadastrado com sucesso!”;
Lembrando que poderíamos também fazer nosso cadastro utilizando os pontos de interrogação.
Exemplo:

```
<?php
    include_once("conexao.php");
```

```
    $nome = "Carlos";
    $sobrenome = "Alberto";


    try{

        $cadastrar = $conexao->prepare("INSERT INTO alunos (nome,sobrenome)
                        VALUES (?,?)";

        $cadastrar->bindValue(1,$nome);
        $cadastrar->bindValue(2,$sobrenome);

        $cadastrar->execute;
        if($cadastrar->rowCount() > 0){

            echo "Aluno cadastrado com sucesso!";
        }else{

            echo "Erro ao cadastrar aluno.";
        }

    }catch(PDOException $error){
        echo "Erro ao cadastrar aluno: ". $error->getMessage();

    }
?>
```

CADASTRANDO COM PDO UTILIZANDO ARRAY

```
<?php
    include_once("conexao.php");

    $nome = "Carlos";
    $sobrenome = "Alberto";


    $dados = array(

    'nome'=> $nome,
    'sobrenome'=> $sobrenome

        );


    try{
        $cadastrar = $conexao->prepare("INSERT INTO alunos (nome,sobrenome)

                        VALUES (:nome,:sobrenome)";
        foreach($dados as $indice => $valor):

            $cadastrar->bindValue(":$indice",$valor);
```

```
        endforeach;

        $cadastrar->execute;
        if($cadastrar->rowCount() > 0){

            echo "Aluno cadastrado com sucesso!";
        }else{

            echo "Erro ao cadastrar aluno.";
        }

    }catch(PDOException $error){
        echo "Erro ao cadastrar aluno: ". $error->getMessage();

    }
?>
```

CADASTRANDO COM PDO UTILIZANDO ARRAY E INTERROGAÇÃO

```
<?php
    include_once("conexao.php");

    $nome = "Carlos";
    $sobrenome = "Alberto";


    $dados = array(

    1=> $nome,
    2=> $sobrenome

        );


    try{
        $cadastrar = $conexao->prepare("INSERT INTO alunos (nome,sobrenome)

                        VALUES (?,?)";
        foreach($dados as $indice => $valor):

            $cadastrar->bindValue($indice,$valor);
        endforeach;

        $cadastrar->execute;
        if($cadastrar->rowCount() > 0){

            echo "Aluno cadastrado com sucesso!";
        }else{

            echo "Erro ao cadastrar aluno.";
```

```
        }

    }catch(PDOException $error){
        echo "Erro ao cadastrar aluno: ". $error->getMessage();

    }
?>
```

Simples, fácil e seguro utilizar o PDO para fazer nosso INSERT né?

Aula 05 – Deletando Dados com PDO

Na aula de hoje iremos aprender a deletar dados do banco de dados utilizando o super PDO =) Primeiramente, como toda aula, vejamos um exemplo utilizando o modo primitivo, ou simples do php, como muitos dizem.

DELETANDO SEM A UTILIZAÇÃO DO PDO

```
<?php
    include_once("conexao.php");

    $matricula = $_GET['matricula'];
    $sql = "DELETE * FROM alunos WHERE matricula = '$matricula'";

    $query = mysql_query($sql);


    if($query){
        echo "Aluno deletado com sucesso!";

    }else{
        echo "Não foi possível deletar o aluno.";

    }
?>
```

Lembrou como era utilizado o modo primitivo do php ? Agora , veja o mesmo utilizando PDO.

DELETANDO UTILIZANDO O PDO

```
<?php
    include_once("conexao.php");

try{
    $matricula = $_GET['matricula'];

    $deletar = $conexao->prepare("DELETE * FROM alunos WHERE matricula =
:matricula");
    $deletar->bindValue(":matricula",$matricula);

    $deletar->execute();
```

```
    if($deletar->rowCount() > 0){
        echo "Aluno deletado com sucesso!";

    }else{
        echo "Não foi possível deletar o aluno.";

    }
}catch(PDOException $error){

    echo "Error ao deletar aluno: ". $error->getMessage();
}

?>
```

LEMBRANDO TAMBÉM, QUE PODERIAMOS UTILIZAR OS FAMOSOS PONTOS DE INTERROGAÇÃO PARA DELETAR.
Exemplo:

```
< ?php
    include_once("conexao.php");

try{
    $matricula = $_GET['matricula'];

    $deletar = $conexao->prepare("DELETE * FROM alunos WHERE matricula = ?");
    $deletar->bindValue(1,$matricula);

    $deletar->execute();


    if($deletar->rowCount() > 0){
        echo "Aluno deletado com sucesso!";

    }else{
        echo "Não foi possível deletar o aluno.";

    }
}catch(PDOException $error){

    echo "Error ao deletar aluno: ". $error->getMessage();
}

?>
```

Simples, e descomplicado.

Aula 06 – Atualizando Dados com PDO

Hoje iremos então, finalizar nosso CRUD (create – read- update – delete = criar – ler – atualizar – deletar).

Na aula de hoje aprenderemos a atualizar os dados utilizando o PDO. Mantendo a rotina exposta em todas as aulas do curso, primeiramente irei mostrar o modo tradicional ou primitivo do php. Logo após, o mesmo exemplo utilizando o PDO.

ATUALIZANDO DADOS SEM UTILIZAR PDO

```
<?php
    include_once("conexao.php");/*Inclui arquivo de conexao com o bd*/

    $matricula = $_POST['matricula'];
    $nome = $_POST['nome'];

    $sobrenome = $_POST['sobrenome'];


$sql = "UPDATE alunos SET
        nome = '$nome',

        sobrenome = '$sobrenome'
        WHERE

        matricula = '$matricula'
        ";

$query = mysql_query($sql);


    if($query){
        echo "Aluno atualizado com sucesso!";

    }else{
        echo "Erro ao atualizar aluno.";

    }
?>
```

ATUALIZANDO DADOS UTILIZANDO PDO

```
<?php
    include_once("conexao.php");/*Inclui arquivo de conexao com o bd*/

    $matricula = $_POST['matricula'];
    $nome = $_POST['nome'];

    $sobrenome = $_POST['sobrenome'];


    $atualizar = $conexao->prepare("UPDATE alunos SET
        nome = :nome,

        sobrenome = :sobrenome
```

```
        WHERE

        matricula = :matricula
              ");

    $atualizar->bindValue(":nome",$nome);
    $atualizar->bindValue(":sobrenome",$sobrenome);

    $atualizar->bindValue(":matricula",$matricula);
    $atualizar->execute;


    if($atualizar->rowCount() > 0){

        echo "Aluno atualizado com sucesso!";
    }else{

        echo "Erro ao atualizar aluno.";
    }

?>
```

PERGUNTA:PODERÍAMOS UTILIZAR OS PONTOS DE INTERROGAÇÃO ? SIM. VEJA:

```
<?php
    include_once("conexao.php");/*Inclui arquivo de conexao com o bd*/

    $matricula = $_POST['matricula'];
    $nome = $_POST['nome'];

    $sobrenome = $_POST['sobrenome'];


    $atualizar = $conexao->prepare("UPDATE alunos SET
        nome = ?,

        sobrenome = ?
        WHERE

        matricula = ?");
    $atualizar->bindValue(1,$nome);

    $atualizar->bindValue(2,$sobrenome);
    $atualizar->bindValue(3,$matricula);

    $atualizar->execute;


    if($atualizar->rowCount() > 0){
```

```
        echo "Aluno atualizado com sucesso!";

    }else{
        echo "Erro ao atualizar aluno.";

    }
?>
```

Pergunta: Poderíamos utilizar arrays para fazer tal atualização? Seria ótimo para realizar múltiplas atualizações de dados correto?
Resposta: Sim, podemos utilizar arrays. E, sim seria ótimo para manipulação múltipla de dados.

EXEMPLO UTILIZANDO ARRAY – SEM A UTILIZAÇÃO DOS PONTOS DE INTERROGAÇÃO.

```
<?php
    include_once("conexao.php");/*Inclui arquivo de conexao com o bd*/

    $matricula = $_POST['matricula'];
    $nome = $_POST['nome'];

    $sobrenome = $_POST['sobrenome'];


    $dados = array(
        'nome'=>$nome,

        'sobrenome'=>$sobrenome,
        'matricula'=>$matricula

            );


    $atualizar = $conexao->prepare("UPDATE alunos SET
        nome = :nome,

        sobrenome = :sobrenome
        WHERE

        matricula = :matricula
                ");

        foreach($dados as $indice => $valor):
            $atualizar->bindValue(":$indice",$valor);

        endforeach;
    $atualizar->execute;


    if($atualizar->rowCount() > 0){

        echo "Aluno atualizado com sucesso!";
```

```
    }else{

        echo "Erro ao atualizar aluno.";
    }

?>
```

EXEMPLO UTILIZANDO ARRAY – UTILIZANDO OS PONTOS DE INTERROGAÇÃO.

```
<?php
    include_once("conexao.php");/*Inclui arquivo de conexao com o bd*/

    $matricula = $_POST['matricula'];
    $nome = $_POST['nome'];

    $sobrenome = $_POST['sobrenome'];


    $dados = array(
        1=>$nome,

        2=>$sobrenome,
        3=>$matricula

            );


    $atualizar = $conexao->prepare("UPDATE alunos SET
        nome = ?,

        sobrenome = ?
        WHERE

        matricula = ?");
        foreach($dados as $indice => $valor):

            $atualizar->bindValue($indice,$valor);
        endforeach;

    $atualizar->execute;


    if($atualizar->rowCount() > 0){
        echo "Aluno atualizado com sucesso!";

    }else{
        echo "Erro ao atualizar aluno.";

    }
?>
```

Viram como é simples fazer atualizações utilizando o PDO?

Desenvolvimento - HTML

HTML Avançado
por Equipe Linha de Código

101 12 2Email4

Tabelas

As tabelas foram uma grande conquista para os autores de documentos para a Web. Com elas é possível, por exemplo, termos estas páginas do tutorial organizadas em colunas, sendo uma delas voltada exclusivamente aos links de navegação e observações. Tabelas implementam um conceito importante de design, as "grades", segundo as quais organizamos textos e ilustrações de maneira organizada.

Como já foi possível perceber, as tabelas podem conter textos, listas, parágrafos, imagens, diversas outras formatações - inclusive outras tabelas. Novas versões de HTML e de navegadores populares vêm acrescentando diversos atributos às tabelas, e nosso objetivo aqui é saber lidar com a maioria desses recursos disponíveis.

A formatação de tabelas faz parte da versão 3.0 da linguagem HTML. Manipular tabelas em HTML dá trabalho e necessita de calma.

A tag para criação de uma tabela é <TABLE>, e dentro dela você coloca atributos importantes para a criação correta da sua tabela. Não esqueça de fazer o fechamento da tag com </TABLE>.

- Atributos de Tabela

- <TABLE BORDER>
O elemento delimita uma tabela e a espessura de sua borda:

<TABLE BORDER=valor da espessura> ... </TABLE>

- WIDTH e HEIGHT
Com estes atributos, você pode definir o tamanho que a tabela vai ocupar na tela. WIDTH (largura), HEIGHT (altura). Eles podem ser em pixels ou em porcentagem. Basicamente o HEIGHT não é usado , mas o valor de largura é muito útil. Veja como usar:

<TABLE BORDER=valor da borda WIDTH="valor da largura">

Um exemplo de tabela com valor de largura que ocupe 75% da tela (WIDTH="75%"):

Coluna 1 Coluna 2

linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2

- Dentro dessa formatação, inserimos as linhas e elementos da tabela:

- <TH> ... </TH>
Define um cabeçalho para a tabela

- <TR> ... </TR>
Delimita uma linha

- <TD> ... </TD>
Delimita um elemento da tabela

Veja como fica a tag para a criação de uma tabela com 2 linhas e 2 colunas:

```
<TABLE BORDER=2>
<TH> Coluna 1</TH><TH> Coluna 2 </TH>
<TR><TD> linha1, coluna 1</td><td> linha 1, coluna 2 </TD></TR>
<TR><TD> linha 2, coluna 1</TD><TD>linha 2, coluna 2 </TD></TR>
</TABLE>
```

Resultado:

Coluna 1 Coluna 2

linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2

Agora veja a mesma tabela sem bordas (border=0):

Coluna 1 Coluna 2
linha1, coluna 1 linha 1, coluna 2
linha 2, coluna 1 linha 2, coluna 2

- Atributos COLSPAN e ROWSPAN

É possível dividir colunas e linhas, através dos atributos COLSPAN (para colunas) e ROWSPAN (para linhas):

```
<TABLE BORDER=2>
<TH COLSPAN=2>Colunas 1 e 2</TH>
<TR>
<TD>linha1, coluna 1</TD><TD> linha 1, coluna 2</TD></TR>
<TR><TD>linha 2, coluna 1</T><TD>linha 2, coluna 2</TD></TR>
<TH ROWSPAN=3>3 linhas juntas</TH>
<TD>uma linha</TD>
<TR><TD>duas linhas</TD></TR>
<TR><TD>três linhas</TD></TR>
</TABLE>
```

Resultado:

Colunas 1 e 2 3 linhas juntas

linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2
uma linha	
duas linhas	
três linhas	

Neste exemplo, temos que o cabecalho Colunas 1 e 2 compreende duas colunas (colspan=2); o cabeçalho 3 linhas compreende, por sua vez, 3 linhas (rowspan=3).

- Cores de fundo

Quando você cria uma tabela, ela tem o fundo transparente e, portanto, vai sempre assumir a cor do fundo ou a imagem de fundo da página. Mas é possível que você defina uma cor de fundo para cada célula, linha e coluna da sua tabela. Para isso você deve incluir o atributo "BGCOLOR", veja como fica:

```
<table border=2>
<th bgcolor="#FF0000">Coluna 1</th>
<th bgcolor="#009900">Coluna 2</th>
<tr>
<td>linha1, coluna 1</td>
<td> linha 1, coluna 2</td>
</tr>
<tr>
<td>linha 2, coluna 1</td>
<td>linha 2, coluna 2</td>
</tr>
</table>
```

Resultado:

Coluna 1 Coluna 2

linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2

Você pode definir a cor de fundo de qualquer linha ou coluna que quiser, bastando apenas incluir o atributo junto da tag de linha ou coluna. Da mesma forma você pode incluir tags de fonte e imagem para colocar dentro de cada célula da tabela.

- Imagem de fundo

Você também pode colocar em uma tabela uma imagem de fundo, como um papel de parede somente para a tabela. Para isso, você só precisa colocar o atributo "BACKGROUND". Veja como fazer:

```
<table border=2 background="nome da imagem que vai aparecer">
<th><font color="#000000">Coluna 1</font></th>
<th><font color="#000000">Coluna 2</font></th>
<tr>
<td><font color="#000000">linha1, coluna 1</font></td>
<td><font color="#000000"> linha 1, coluna 2</font></td>
</tr>
<tr>
<td><font color="#000000">linha 2, coluna 1</font></td>
<td><font color="#000000">linha 2, coluna 2</font></td>
</tr>
</table>
```

Resultado:

Coluna 1Coluna 2

linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2

Assim como quando colocamos uma imagem de fundo em uma página, a imagem de fundo em uma tabela será "repetida" até preencher todo o espaço da tabela.

Onde houver uma cor de fundo definida para a célula, coluna ou linha a imagem de fundo não vai aparecer. Veja:

Coluna 1Coluna 2

linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2

Frames - Parte 1

Os frames são divisões da tela do navegador em diversas telas (ou "quadros"). Com isso, torna-se possível apresentar mais de uma página por vez, por exemplo, um índice principal em uma parte pequena da tela, e os textos relacionados ao índice em outra parte. Trata-se de um recurso muito usado para criar menus de navegação em que o menu sempre está fixo e somente o outro frame muda para que apareçam as páginas.

Para criar uma página com 2 frames, você terá que criar no mínimo 3 páginas, para resultar em uma. Isso porque, neste caso, cada frame é uma página HTML (2 frames) e há uma terceira página principal que vai ter as informações destes frames dentro dela.

Para criar um frame, primeiro crie duas páginas simples e salve no mesmo diretório. Crie agora uma terceira página, com a seguinte tag:

```
<HTML>
<HEAD>
<TITLE> Coloque o Título aqui </TITLE>
</HEAD>
<FRAMESET COLS=20%,80%>
<FRAME SRC="a.html">
<FRAME SRC="b.html">
</FRAMESET>
<BODY>
</BODY>
</HTML>
```

Lembre-se de sempre colocar o código para os frames abaixo da tag </head> e antes da tag <body>.

A parte FRAMESET define a divisão da página em "quadros". Neste exemplo, a página será dividida em duas colunas (Atributo COLS), sendo a primeira com 20% do tamanho da tela, e a segunda coluna com os restantes 80% da tela.

Dentro da formatação de FRAMESET, temos os FRAME SRC, que são referências às páginas que serão mostradas nos frames definidos Assim, aqui vemos que a página "a.html" será mostrada na primeira coluna (que ocupará 20% da tela), e "b.html" será mostrada na segunda (ocupando 80% da tela).

Substitua "a.htm" para a primeira página que você criou por "b.htm" na segunda.
Salve e veja o resultado.

Este é o comando básico para fazer uma página com frames. Vamos ver os tipos de frames que é possível

você criar. Na seção de "Download de Exemplos" você poderá baixar arquivos com exemplos de frames para facilitar a construção de suas páginas.**Frames - Parte 2**

Como já foi possível observar, FRAMESET tem atributos que definem a divisão da janela do navegador em colunas; essa divisão também pode ser feita em linhas, e utilizando uma combinação de "framesets" para variadas apresentações. Lembre-se de sempre colocar o código para os frames abaixo da tag </head> e antes da tag <body>. Confira os tipos mais comuns de divisões que você pode fazer. Você poderá fazer o download dos arquivos de exemplos neste tutorial.

```
<FRAMESET COLS="20%, 80%"> <FRAME SRC="coluna1.html"> <FRAME SRC="coluna2.html"> </FRAMESET>
```

```
<FRAMESET ROWS="20%, 80%"> <FRAME SRC="linha1.html"> <FRAME SRC="linha2.html"> </FRAMESET>
```

```
<FRAMESET COLS="20%, 80%"> <FRAME SRC="coluna1.html"> <FRAMESET ROWS="20%, 80%"> <FRAME SRC="linha1.html"> <FRAME SRC="linha2.html">
</FRAMESET> </FRAMESET>
```

```
<FRAMESET ROWS="20%, 80%"> <FRAME SRC="linha1.html"> <FRAMESET COLS="20%, 80%"> <FRAME SRC="coluna1.html"> <FRAME SRC="coluna2.html">
</FRAMESET> </FRAMESET>
```

Lembre-se de que os frames fixos não precisam ter nomes, mas os frames que receberão textos, sim!

O exemplo simples visto até agora segue o modelo da primeira composição (em colunas). As composições com mais de um frameset precisam ser bem planejadas para funcionarem bem. Os valores dos atributos COLS e ROWS, você pode alterar conforme a necessidade.

- Atributos Especiais:

Até este ponto, vimos os atributos COLS e ROWS (para FRAMESET), SRC e NAME (para FRAME). Outros atributos permitem um maior controle sobre a apresentação:

- FRAMEBORDER="NO"
Eliminação das bordas dos frames: "NO" sem bordas, "YES" com bordas.

- BORDER="valor do espaço"
Eliminação do espaço entre os frames.

- SCROLLING="NO"
Frame sem barra de rolagem. A opção "YES" define a existência da barra de rolagem.

Combinando estes atributos em uma tag de frames simples teríamos:

```
<FRAMESET COLS="20%, 80%" FRAMEBORDER="NO" BORDER="0" SCROLLING="NO">
<FRAME SRC="col1.html">
<FRAME SRC="col2.html">
</FRAMESET>
```

Frames - Parte 3

Sempre que se aciona um link dentro de uma página, o padrão é que a página referente a esse link seja carregada na mesma janela da página anterior. No exemplo visto com frames, seguir um link dentro da

janela à direita faz com que a página apontada seja mostrada ocupando a janela da direita (coluna de 80%). Para isso você tem que indicar ao navegador, em que frame deseja que a página do link seja mostrada. Para isso você usa o atributo "NAME".

```
<HTML>
<HEAD>
<TITLE>Coloque o Título aqui</TITLE>
</HEAD>
<FRAMESET COLS="20%, 80%">
<FRAME SRC="coluna1.html">
<FRAME SRC="coluna2.html" NAME=principal>
</FRAMESET>
</HTML>
```

Veja no código-fonte acima que o frame associado à coluna2.html tem um atributo NAME. Isso faz com que se possa "direcionar" o frame em que queremos mostrar determinada página ao acionarmos um link. Veja como ficaria a tag de link para isso:

```
<a href="apresenta2.html" target="principal">Exemplo</a>.
```

Quando se acionar esse link, a página será mostrada no frame onde está a coluna2.html, ou seja, em vez de carregar o link na mesma coluna em que está coluna1.html, ela será mostrada na coluna em que está coluna2.html. O "Target" define que o link deve abrir no frame que tem este nome, no caso o "coluna2.html". Você define o nome de cada frame e, no target, você define para qual frame será aberto o link.

Download de Exemplos

Agora que você já viu como criar os frames e alterar seus atributos, você poderá fazer o download de arquivos de exemplo que vão servir para que você possa ter uma idéia e uma base para o início da construção de suas páginas com frames. Ao lado de cada exemplo você encontra o link para fazer o download do arquivo que está no formato "exe". Todos devem ser colocados em um mesmo diretório. Abrindo o arquivo "principal.html" você vê o exemplo, os outros arquivos são os frames que você poderá editar com tudo aquilo que você está aprendendo aqui.

Exemplo 1:

Fazer o download deste exemplo

Exemplo 2:

Fazer o download deste exemplo

Exemplo 3:

Fazer o download deste exemplo

Exemplo 4:

Fazer o download deste exemplo

Formulários - Parte 1

Um formulário é um modelo para a entrada de um conjunto de dados. Eles são muito usados em registros eletrônicos ou em formulários para o envio de emails para as páginas sem precisar abrir um programa de

email ou digitar endereços, além da vantagem de ter campos para cada tipo de informação.

O primeiro passo para fazer formulários é aprender as tags que fazem os campos de entrada de dados.

O elemento <FORM> inicia um formulário e contém uma seqüência de elementos de entrada e de formatação do documento.

<FORM ACTION="URL_de_script" METHOD=método>...</FORM>

- Atributos da tag <FORM>

- ACTION
Especifica o URL do script ao qual serão enviados os dados do formulário.

- METHOD
Seleciona um método para acessar o URL de ação. Os métodos usados atualmente são GET e POST. Ambos os métodos transferem dados do navegador para o servidor, com a seguinte diferença básica:

. POST
Os dados inseridos fazem parte do corpo da mensagem enviada para o servidor; transfere grande quantidade de dados.
. GET
Os dados inseridos fazem parte do URL associado à consulta enviada para o servidor; suporta até 128 caracteres.

Os formulários podem conter qualquer formatação - parágrafos, listas, tabelas, imagens - exceto outros formulários. Em especial, colocamos dentro da marcação de <FORM> as formatações para campos de entrada de dados, que são três: <INPUT>, <SELECT> e <TEXTAREA>.

Todos os campos de entrada de dados têm um atributo NAME, ao qual associamos um nome, utilizado posteriormente pelo sistema para enviar os dados. Normalmente são usados "scripts".
Os scripts que organizam esses dados de entrada de todos os campos em um conjunto de informações e realizem uma tarefa programada, como por exemplo enviar os dados do formulário para o seu email. HTML não tem condições de fazer isso, por isso é necessário utilizar scripts CGI, PERL ou ASP para executar estas tarefas, porém estes tipos de scripts necessitam de aprendizado mais dedicado para se criar o que você deseja e são muito mais complexos do que a linguagem HTML, pois eles processam informações.

Você pode usar scripts prontos em seus formulários, mas no momento ainda não é possível membros da StarMedia Home Pages executarem os scripts nos servidores.**Formulários - Parte 2**

Agora precisamos criar os campos onde os usuários vão digitar os dados e selecionar as opções. Você inclui estes campos dentro da tag <FORM>. Conheça a função de cada um e como criá-los:

<INPUT>
Esta é a tag para iniciar a criação de campos de dados. Há vários atributos que permitem a criação de diferentes campos de entrada de dados. Vejamos:

- Campo de dados texto

Quando INPUT não apresenta atributos, assume-se TYPE=TEXT como padrão à formatação:

<FORM>
Nome: <INPUT TYPE=TEXT NAME="Nome">
</FORM>

Ou apenas:

```
<FORM>
Nome: <INPUT NAME="Nome">
</FORM>
```

Resultado:

Nome:
- Campo de dados de texto em formato senha

Entrada de texto na qual os caracteres são escondidos por asteriscos. É muito usado para entradas de senhas, como se pode ver no exemplo.

```
<FORM>
Usuário: <INPUT TYPE=TEXT NAME=login><br>
Senha: <INPUT TYPE=PASSWORD NAME="senha">
</FORM>
```

Resultado:

Usuário:
Senha:
- Atributo NAME

Ele é especialmente para que você dê um nome ao campo, ele não aparece na página, mas serve para identificar o campo e o valor digitado no email que você receber, nunca deixe de definir o nome dos campos, só assim você você poderá saber o que cada usuário preencheu em cada campo. Por exemplo se você tem vários campos de texto, cada um para um tipo de infromação diferente, você usa o name para identificar o campo. Você vai ver sempre que em todas as tags INPUT este atributo está presente. Basta incluir ele da seguinte forma:

```
<INPUT TYPE=TEXT NAME="coloque o nome do campo">
```

- Atributo VALUE

- VALUE
Pode ser usado para dar um valor inicial a um campo de tipo texto ou senha. Desse modo, se o usuário não preencher este campo, será adotado este valor padrão. Se o usuário quiser inserir dados, ele somente precisará apagar o que já estiver escrito.

```
Nome: <INPUT TYPE=TEXT NAME="nome" VALUE="Seu nome">
```

Resultado:

Nome:

- Atributo SIZE

- SIZE
Especifica o tamanho do espaço no vídeo para o campo do formulário. Só é válido para campos TEXT e PASSWORD. O valor padrão é 20.

```
<FORM>
Endereço: <INPUT TYPE=TEXT SIZE=35>
```

</FORM>

Endereço:
- Atributo MAXLENGTH

- MAXLENGTH
É o número de caracteres aceitos em um campo de dados; este atributo só é válido para campos de entrada TEXT e PASSWORD.

<FORM>Dia do mês:
<INPUT TYPE=TEXT NAME="ex" MAXLENGTH=2>
</FORM>

Resultado:

Dia do mês:
Apenas 2 caracteres serão lidos pelo formulário, não importa o quanto se escreva neste campo.

- Campo de Dados Escondido (Hidden)

Este tipo de campo funciona igual a um campo de texto, só que ele não aparece no formulário para o visitante. Ele está lá no código, mas o visitante não pode vê-lo ou alterá-lo. Isso é importante para você incluir informações que achar necessárias, mas que não deseja que o visitante altere.

Veja um exemplo:

<FORM>
<INPUT TYPE=HIDDEN NAME=Escondido Value=Sim>
</FORM>

Aqui o campo está escondido. O visitante não o vê, mas ele vai ser processado pelo formulário. Você pode incluí-lo sem problemas junto com os outros elementos. Por exemplo:

<FORM>
Usuário: <INPUT TYPE=TEXT NAME=login>
Senha: <INPUT TYPE=PASSWORD NAME="senha">
<INPUT TYPE=HIDDEN NAME=Escondido Value=Sim>
</FORM>

- Múltipla escolha

Você também pode criar um campo onde o usuário só precisa selecionar as opções, sem digitar nada. Há dois tipos básicos de seleção:

- CHECKBOX
Insere um botão de escolha de valores para várias opções.

<INPUT TYPE=CHECKBOX NAME="esporte" VALUE="basquete">Basquete
<INPUT TYPE=CHECKBOX NAME=esporte VALUE=bocha>Bocha
Um atributo "CHECKED" marca uma escolha inicial, isto é, se o usuário não escolher nenhuma opção, o formulário irá considerar a alternativa "pré-escolhida":

<INPUT TYPE=CHECKBOX NAME="esporte" VALUE="volei" CHECKED&>Vôlei

<INPUT TYPE=CHECKBOX NAME="esporte" VALUE="futebol">Futebol

Resultado:

Esportes que você pratica:
Futebol
Vôlei
Natação
Basquete
Tênis
Bocha

- Escolha única

- RADIO
Insere um botão de escolha de valores para uma opção, isto é, somente uma alternativa pode ser escolhida.

<INPUT TYPE=RADIO NAME="time" VALUE="palm">Palmeiras
<INPUT TYPE=RADIO NAME="time" VALUE="inte">Internacional
Uma diretiva CHECKED marca uma escolha inicial - se o usuário não escolher nenhuma opção, o formulário irá considerar a alternativa "pré-escolhida":

<INPUT TYPE=RADIO NAME="time" VALUE="aea" CHECKED>Cruzeiro
<INPUT TYPE=RADIO NAME="time" VALUE="naut">Náutico

Resultado:

Seu time do coração:
Palmeiras Internacional Cruzeiro

- Botões de ação
Os botões são usados para executar ações dentro do formulário, como enviar os dados ou limpar os campos.

- SUBMIT

Apresenta o botão que causa o envio dos dados de entrada para o servidor.

<FORM>
<INPUT TYPE=SUBMIT>
</FORM>

Resultado:

É possível modificar a mensagem desse botão através do atributo VALUE.

<FORM>
<INPUT TYPE=SUBMIT VALUE="Envia mensagem">
</FORM>

- RESET

Restaura os valores iniciais das entradas de dados.

```
<FORM>
<INPUT TYPE=RESET>
</FORM>
```

Resultado:

É possível modificar a mensagem desse botão através do atributo VALUE

```
<FORM>
<INPUT TYPE=RESET VALUE="Apaga tudo!">
</FORM>
```

Formulários - Parte 3

- <SELECT>

Apresenta uma lista de valores em um menu suspenso. Ótimo para criar escolhas que o usuário seleciona sem precisar digitar através de campos <OPTION>. Veja como fazer: .

```
<SELECT NAME="sabor">
<OPTION>Abacaxi
<OPTION>Creme
<OPTION>Morango
<OPTION>Chocolate
</SELECT>
```

Resultado:

Abacaxi Creme Morango Chocolate

- Atributo SELECTED

Também é possível estabelecer uma escolha-padrão, através do atributo SELECTED

```
<SELECT NAME="sabor">
<OPTION>Abacaxi
<OPTION SELECTED>Creme
<OPTION>Morango
<OPTION>Chocolate
</SELECT>
```

Resultado:

Abacaxi Creme Morango Chocolate Aqui o Creme já aparece previamente selecionado.

- Atributo SIZE

Com o atributo SIZE, escolhe-se quantos itens da lista serão mostrados (no exemplo, SIZE=4):

Abacaxi Creme Morango Chocolate - <TEXTAREA>

Abre uma área para entrada de texto, de acordo com atributos para número de colunas, linhas, e - se for o caso - um valor inicial. É um campo onde as pessoas poderão digitar mais texto do que nos campos de texto que vimos na parte 2. Além disso, o usuário pode ver o texto que está digitando. A tag fica assim:

<TEXTAREA COLS=40 ROWS=5 NAME="comentario"> Deixe seu comentário </TEXTAREA>

Resultado:

Deixe seu comentário

Repare que, no atributo "COLS", definimos o número de colunas para a largura do campo de texto e, em "ROWS", o número de linhas para o campo de texto. Se o usuário digitar mais do que cinco linhas, surgirá uma barra de rolagem para ele cortar o texto se desejar.

Os valores destes atributos podem ser modificados à vontade de acordo com a sua necessidade.**Inserindo Sons**

Você pode incluir em sua página um som de fundo. Com isso, sempre que alguém acessar sua página, este som será tocado. Desta forma você pode fazer seu site ter mais recursos e personalidade.

Os arquivos de som ideais para serem inseridos em uma home page são aqueles no formato MIDI (.MID), mas você também pode incluir sons do tipo WAVE (.WAV).

O Netscape e o Internet Explorer têm algumas diferenças em seu sistema e, por isso, em algumas versões do Netscape o som pode não tocar.

A tag deve ser <EMBED>. Veja o exemplo:

<embed src="nome da música" autostart="true" loop=10>

- Em "nome da musica" você coloca o nome da música mid ou wav.
- Em "loop" você coloca o número de vezes que o som vai ser repetido.
- Em "autostart" você define se quer que a música comece a tocar sozinha (atributo TRUE), se você colocar "FALSE" o usuário terá que clicar no botão de "play"para ouvir a música

Se você quiser que ele fique se repetindo infinitamente, basta colocar "infinity" no lugar de um número. Não é necessário colocar "</embed>" neste caso.

Âncoras

Âncoras são links estabelecidos para certas partes de uma mesma página ou de outra página qualquer, por exemplo, se você quer que um link vá direto para um texto que está no meio de uma página sem o visitante ter que descer a página. Por exemplo, clicando aqui você salta automaticamente para o fim da página.

Para o funcionamento de uma âncora deve-se estabelecer dois comandos:
- Um que defina o lugar da página para onde se pretende "saltar" e
- Outro que identifique esse ponto de destino. - Links com âncoras para uma mesma página
Você pode fazer com que links saltem para determinados pontos de uma mesma página, facilitando a navegação. Veja como. Primeiro vamos definir o ponto de destino. Você deve incluir a seguinte tag exatamente no local onde quer que o link vá:

<A NAME="Coloque o nome da ancora aqui"></A>

- NAME é um parâmetro que define um ponto de destino.

- Agora você pode usar a tag de link "comum" para fazer o link saltar para o ponto onde você definiu:

<a href="#nome da ancora">Texto</a>

Repare que há um símbolo "#" antes do nome da âncora. É ele que vai fazer o navegador entender que deve ir para a âncora que você definiu.

- Links com âncoras para uma outra página

Agora digamos que você queira que o link vá para uma outra página e direto para um texto no final dela. Primeiro você deve incluir, na página de destino, a tag da âncora no ponto onde ela vai aparecer ela depois de clicar no link, e salvar a página. Em seguida, na página que terá o link, você deve construir a tag assim:

<a href="nome-do-lugar-a-ser-levado#nome da ancora">Texto</a>

Em "nome-do-lugar-a-ser-levado" você coloca o URL ou a página html que vai ser mostrada e, depois do destino, o "#nome da ancora". Este nome de âncora deve ser o mesmo nome de âncora que você definiu na página de destino.

Você pode incluir quantas âncoras quiser em sua página, mas lembre-se de que cada uma deve ter um nome diferente.

Cores Hexadecimais

A tabela abaixo apresenta os códigos de cores que podem ser utilizados em documentos HTML.
As cores marcadas por um asterisco vermelho podem ser escritas no lugar dos códigos.

Nome da cor	Código da cor	Aparência
* White	#FFFFFF	
* Red	#FF0000	
* Green	#00FF00	
* Blue	#0000FF	
Magenta	#FF00FF	
Cyan	#00FFFF	
* Yellow	#FFFF00	
* Black	#000000	
* Aqua	#70DB93	
Baker"s Chocolate	#5C3317	
Blue Violet	#9F5F9F	
Brass	#B5A642	
Bright Gold	#D9D919	
Brown	#A62A2A	
Bronze	#8C7853	
Bronze II	#A67D3D	
Cadet Blue	#5F9F9F	
Cool Copper	#D98719	
Copper	#B87333	
Coral	#FF7F00	
Corn Flower Blue	#42426F	
Dark Brown	#5C4033	
Dark Green	#2F4F2F	

Dark Green Copper	#4A766E	
Dark Olive Green	#4F4F2F	
Dark Orchid	#9932CD	
Dark Purple	#871F78	
Dark Slate Blue	#6B238E	
Dark Slate Grey	#2F4F4F	
Dark Tan	#97694F	
Dark Turquoise	#7093DB	
Dark Wood	#855E42	
Dim Grey	#545454	
Dusty Rose	#856363	
Feldspar	#D19275	
Firebrick	#8E2323	
Forest Green	#238E23	
Gold	#CD7F32	
Goldenrod	#DBDB70	
*Gray	#C0C0C0	
Green Copper	#527F76	
Green Yellow	#93DB70	
Hunter Green	#215E21	
Indian Red	#4E2F2F	
Khaki	#9F9F5F	
Light Blue	#C0D9D9	
Light Grey	#A8A8A8	
Light Steel Blue	#8F8FBD	
Light Wood	#E9C2A6	
*Lime	#32CD32	
Mandarian Orange	#E47833	
*Maroon	#8E236B	
Medium Aquamarine	#32CD99	
Medium Blue	#3232CD	
Medium Forest Green	#6B8E23	
Medium Goldenrod	#EAEAAE	
Medium Orchid	#9370DB	
Medium Sea Green	#426F42	
Medium Slate Blue	#7F00FF	
Medium Spring Green	#7FFF00	
Medium Turquoise	#70DBDB	
Medium Violet Red	#DB7093	
Medium Wood	#A68064	
Midnight Blue	#2F2F4F	
* Navy	#23238E	
Neon Blue	#4D4DFF	
Neon Pink	#FF6EC7	
New Midnight Blue	#00009C	

New Tan	#EBC79E	
Old Gold	#CFB53B	
Orange	#FF7F00	
Orange Red	#FF2400	
Orchid	#DB70DB	
Pale Green	#8FBC8F	
Pink	#BC8F8F	
Plum	#EAADEA	
Quartz	#D9D9F3	
Rich Blue	#5959AB	
Salmon	#6F4242	
Scarlet	#8C1717	
Sea Green	#238E68	
Semi-Sweet Chocolate	#6B4226	
Sienna	#8E6B23	
* Silver	#E6E8FA	
Sky Blue	#3299CC	
Slate Blue	#007FFF	
Spicy Pink	#FF1CAE	
Spring Green	#00FF7F	
Steel Blue	#236B8E	
Summer Sky	#38B0DE	
Tan	#DB9370	
Thistle	#D8BFD8	
Turquoise	#ADEAEA	
Very Dark Brown	#5C4033	
Very Light Grey	#CDCDCD	
Violet	#4F2F4F	
Violet Red	#CC3299	
Wheat	#D8D8BF	
Yellow Green	#99CC32	